بيتا
1

ANDROID

ببساطة

تعلم تطوير التطبيقات على أجهزة أندرويد



تأليف

محمد بدوي

وائل علواني

أندرويد ببساطة

تعلم تطوير تطبيقات على نظام تشغيل أندرويد المفتوح المصدر

انتهى في (أبريل / 2010)

نشر في (مارس / 2011)

بواسطة

محمد بدوي

وائل علواني

فهرس المحتويات

حول الكتاب

من الجيد معرفته أننا قمنا بكتابة الكتاب في نهاية العام (2009) ميلادية و من ثم حاولنا البحث عن راعي لطباعة الكتاب أو توزيعه و قد قامت بعض الجهات بإطلاق الوعود الكثيرة ومنها شركة جوجل ممثلة بوكيلها في السعودية مما جعلنا الانتظار بضعة أشهر لنعرف رأي الراعي بالكتاب و مدى جودته لتوفيره للناس ، و لكن و مع مرور الوقت لم نعد نسمع منهم ، و لذا تقاعسنا في نشر الكتاب و لم نكمله و توقف العمل عليه نهائيا منذ شهر (4) للعام (2010) و ها قد قمنا أخيرا بتنقيحه و على عجل لإصداره للناس و هو و إن كانت بعض البيانات قديمة إلا أنه مازال صالحا للعمل بما فيه حتى يومنا هذا ،لذا في حال وجود بعض الأخطاء فأرجوا أن تعذرنا فلقد قمنا بأفضل ما يمكن للخروج بالكتاب بأفضل شكل و بأبسط ما يمكن حيث لا يوجد أي كتب عربية حتى الآن مهتمة في هذا المجال و كذلك الحال أغلب الكتب الأجنبية التي طالعناها تعاني من التعقيد ، بل و تعتبر بأن من يقرأ الكتاب ملم بالكثير من الأمور و لذا مازلنا نجد بأن الكتاب سيكون من أفضل ما كتب حتى يومنا هذا في هذا العالم.

الكتاب مجاني للأفراد و يمكن إعادة توزيعه و نشره كما هو بدون أي تعديل و بشكل الكتروني دون الرجوع لنا ، ولذا أرجوا أن لا تنسونا من دعوة بظهر الغيب فنحن بأشد الحاجة لها دائما.

أما في حال الطبع أو استخدامه لأغراض تجارية أو لغير الأفراد فيرجى الرجوع لنا لأخذ إذن خاص بذلك.

حول المؤلفين

محمد بدوي (مدون ،مبرمج ،رائد أعمال)

مطور للتطبيقات على الأجهزة الذكية مثل (iPhone/iPad) و (Android) و (Windows Mobile) و (Blackberry)

الموقع على الانترنت: www.badwi.com
البريد الالكتروني: badwisoft@gmail.com
الهاتف: 00966503415343
حساب على تويتر: http://twitter.com/badwi
حساب فيس بوك: http://www.facebook.com/badwi

وائل علواني (مدون ومطور تطبيقات أندرويد)

مهتم بالتقنية وتطوراتها وخصوصا مجال الأندرويد ومواضيع تعدين البيانات Data Mining.. يحمل ماجستير في علوم الحاسب - تخصص ذكاء صنعي من جامعة الملك عبدالله للعلوم والتقنية (كاوست).

البريد الالكتروني: wael.alalwani@gmail.com
حساب على تويتر: http://twitter.com/waelalwani

لنتعرف بالبداية على أندرويد

أندرويد هو نظام تشغيل مفتوح المصدر موجهة للأجهزة المحمولة اشترته شركة جوجل من أحد الشركات في العام (2005) ميلادية و في العام (2007). تم تبنيه تحت اتحاد المصادر المفتوحة للأجهزة المحمولة (Open Handset Alliance) (OHA) و الذي ضم أكثر من (48) شركة من كبار الشركات و منهم مصنعين و شركات تطبيقات برمجية و شركات تزود شرائح الكترونية و شركات تهتم بالمحتوى و مزودي خدمة الاتصالات. و في العام (2008) قامت شركة (HTC) و المصنعة للأجهزة المحمولة بإصدار أول هاتف متنقل بنظام تشغيل (Android) و في العام (2009) قامت شركة (T-Mobile) ببيع أول خطوط هاتف ضمن عقد سنوي مع أجهزة (HTC) و المتضمنة لنظام تشغيل (أندرويد) .

نظام أندرويد يعتمد ومبني على نواة لينكس (Linux 2.6 kernel) و لكنه لا يعد نظام تشغيل (GNU/Linux) حيث أنه لا يدعم كل من (native windowing system) و (glibc support) و لا حتى أدوات (GNU/Linux) و لكنه يستفيد من النواة بقواتها في الأمان (security) و إدارة الذاكرة (memory management) و إدارة العمليات الخلفية (process management) و كذلك الحال في إدارة الشبكة (network stack) و نموذج السواقات (driver model) و طبقة التجريد (abstraction layer) .

يمكن بناء التطبيقات و تطوير التطبيقات الموجهة لنظام تشغيل (أندرويد) بواسطة كل من (Java) و (Managed Code) و لكن

لا تعبر نسخة (Java) القياسية بل هي مطورة لجزء من (Java5 SE) و تعمل على (JDV) اختصارا لـ (Java Dalvik Virtual) بدلا من (JVM) اختصارا لـ (Java Virtual Machine).

أندرويد و ببساطة هو نظام تشغيل مفتوح المصدر ويدعم تعدد التطبيقات. أي من الممكن أن يعمل به أكثر من تطبيق في نفس الوقت أو الخدمات في الخلفية. و تعمل كل التطبيقات فيه بشكل متساوي من ناحية مشاركة موارد الجهاز وان كان كل تطبيق يعمل بشكل مستقل بطبقة منفصلة. و تمكنك منصته من إعادة استخدام كائناته كما و يستخدم محرك (Web Kit OSS) كمتصفح انترنت و الذي يستخدم في متصفح (Google chrome) . كما يدعم مكتبة (OpenGL ES) لإدارة الصور الثلاثية و يعتمد قواعد بيانات (SQLite) في تخزين البيانات . كما و يدعم كل الصيغ القياسية لوسائط الميديا من صور و أصوات و فيديو و الجميل فيه أنه يتضمن عتاد متنوع من الكاميرا الرقمية و نظام التوضع العالمي (GPS) و البوصلة بالإضافة إلى دعم خاصية الإحساس بالمكان (accelerometer) .

لنتعرف أكثر على بنية أندرويد الداخلية. .

في الأساس تقع نواة لينوكس (Linux Kernel) و التي هي مسئولة عن:

- **Display Driver**
- **Bluetooth Driver**
- **Camera Driver**

- **Flash Memory Driver**
- **Binder (IPC) Driver**
- **Keypad Driver**
- **USB Driver**
- **Wi-Fi Driver**
- **Audio Driver**
- **Power Management**

ثم تليها طبقة التشغيل و التي هي عبارة عن :

- **Dalvik Virtual Machine**
- **Core Libraries**

و هذه المكتبات هي :

- **Surface Manager**
- **Media Framework**
- **SQLite**
- **OpenGL ES**
- **Free Type**
- **Web Kit**
- **SGL**
- **SSL**
- **Libbc**

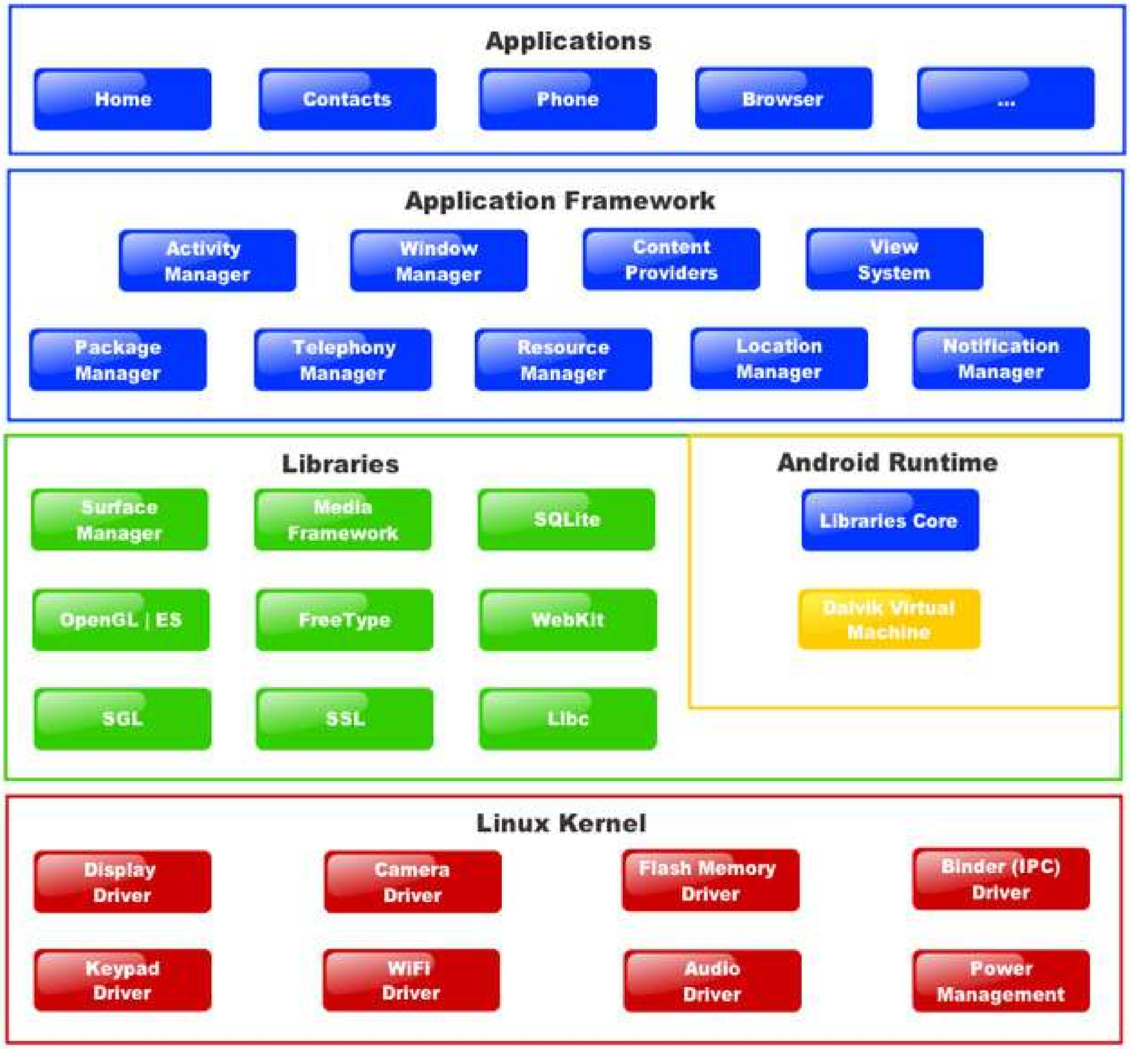
ثم تلي هذه الطبقة طبقة منصة التطبيقات و التي تستخدم من قبل كل التطبيقات:

- **Activity Manager**
- **Window Manager**
- **Content Provider**
- **View System**
- **Notification Manager**
- **Package Manager**
- **Telephony Manager**
- **Resource**
- **Location Manager**
- **Sensor Manager**

ثم في الطبقة الأخيرة تأتي التطبيقات الرئيسية المشحونة من الشركة مثل :

- **Home**
- **Contacts**
- **Phone**
- **Browser**

و بعض التطبيقات الأخرى و التي سيكون منها تطبيقك الخاص الذي سوف نتعلم سوية كيف ستقوم ببنائه.



مصدر الصورة: صفحة أندرويد-ويكيبيديا

الوحدات الأساس الذي سنتعامل معه لأي تطبيق أندرويد هو:

- **Activity**
- **Intent**
- **Service**
- **Content Provider**

قبل الإبحار دعنا نتعرف على هذه الأسس و التي يمكن لتطبيقك أن يستخدم أحدها فقط أو أكثر أو كلها مجتمعة:

النشاط أو الجلسة (Activity)

تستخدم لتوصيف شاشة وحيدة و تكون عبارة عن (Class) و التي يتم استدعاؤها عن طريق الزناد (Intent) و تكون بالعادة الجزء الظاهر إذ يمكن أن تكون في أعلى كل التطبيقات و يمكن استدعائها أو جعلها في الخلفية و إعادة استدعائه أو إغلاقها نهائيا .

دورة حياة (Activity)

بداية (Start) النشاط (Activity) و التي تتعامل مع الأحداث

- **On Create**
- **On Start**
- **Om Restore Instance State**
- **On Resume**

أما في المرحلة الثانية و هي عندما يكون النشاط في وضع التشغيل (Running) فإنه يتعامل مع الأحداث التالية:

- **On Start**
- **On Restart**
- **On Resume**
- **On Push**
- **On Save Instance State**

و بينما النشاط يعمل و الانتقال إلى نشاط أخر فإنه يكون في وضعية الإيقاف المؤقت (Paused) و الذي يمكن التعامل معه مع الأحداث:

- **On Save Instance State**
- **On Resume**
- **On Stop**

أما المرحلة الأخيرة و التي تعتبر نهاية النشاط و التي يتم إلغاؤه من الذاكرة (Destroyed) و يتم تنفيذ الحدث الوحيد:

- **On Destroy**

الزناد / الحدث (Intent)

و الذي يترجم إلى قصد أو غرض و لكن أرى أن زناد أقرب ترجمة للفهم باللغة العربية و التي يستخدم لإتمام عمل ما ، مثلا في استدعاء نشاط جديد مثل شاشة أو عند استدعاء صفحة انترنت أو البحث عن عنصر في قائمة الاتصالات و غيرها من الأمور التي تستخدم بناء على طلب المستخدم.

الخدمة (Service)

تستخدم لتعمل وتبقى كذلك في الخلفية مثلا و بالعادة لا تحتوي على واجهة مستخدم فعلى سبيل المثال لو جربت و قمت بتشغيل مشغل الموسيقى (Media Player) فيمكنك الخروج منه و تصفح بريدك أو كتابة ملاحظاتك و في نفس الوقت مازلت تسمع الموسيقى و في هذه الحالة فإن الموسيقية تعمل كخدمة بالخلفية كـ (Service).

تزويد و تبادل المحتوى (Content Provider)

يمكنك عبر ذلك مناداة أي محتوى خارجي مثلا عبر تطبيق ويب أو حتى في مبادلة المحتوى بين التطبيقات فعلى سبيل المثال يمكنك أن تصل لقائمة عناوين الاتصال في تطبيق (Contacts).

أشكال النشاط (Activity)

يكون عبارة عن نافذة (Window) متعددة فقد فتكون صغيرة أو مليء الشاشة و لتوضع الكائنات عليها فإنها تتعامل بأربع وضعيات مختلفة

عرض بشكل سطري (Linear Layout)

تعرض الكائنات بشكل سطري أي كل عنصر يقع بأسفل العنصر الذي يليه، أو بمحاذاته بحسب الوضع المختار (vertical or horizontal).

بشكل نموذج (Frame Layout)

تعرض الكائنات بشكل متتالي ابتداء من الجهة العليا باليسار.

بشكل علائقي (Relative Layout)

تعرض الكائنات بشكل علائقي فيما بينها أي مثلا أن تضع زر بعد العنصر الفلاني و قبل العنصر الفلاني أو لتملئ مساحة ما بين عنصر (A) و عنصر (B).

بشكل جدولي (Table Layout)

تعرض الكائنات بشكل جدولي حيث يمكن وضع كل كائن داخل خلية كما الحال في تصميم الجداول في (HTML).

قاموس المصطلحات

في هذا القسم سوف نتعرف على بعض المسميات و الاختصارات و المصطلحات التي ستمر علينا في الكتاب و لنقوم بتعريفها هنا لكي نستطيع المضي قدما أثناء شرح باقي أقسام الكتاب لكي لا نحتاج إلى تعريفها في كل مرة نقوم بذكرها

SDK (Software Development Kit)
الحزمة التطويرية البرمجية والتي تحوي المجموعات\كلاسات والدوال التي يمكن استخدامها في تطوير تطبيقات الأندرويد.
NDA (Native Development Kit)
حزمة التطوير بالكود الأصلي
ADT (Android Development Tools)
أدوات تطوير أندرويد
JDK (Java Development Kit)
أدوات تطوير جافا
JRE (Java Runtime Environment)
بيئة تشغيل جافا
JVM (Java Virtual Machine)
آلة جافا الافتراضية
AVD (Android Virtual Device)
محاكي جهاز أندرويد
AVDM (Android Virtual Device Manager)
مدير محاكي أجهزة أندرويد
APK (Android Package)
حزمة أندرويد و هو الملف التنفيذي النهائي للتطبيق

DDMS (Dalvik Debug Monitor Server)
أداة مراقبة محاكي الجهاز
GPS (Global Position System)
نظام التوضع العالمي عبر الأقمار الصناعية
CMD (Command Line)
سطر الأوامر
ADB (Android Debug Bridge)
جسر التواصل مع أجهزة أندرويد لمراقبة الأداء و معالجة الأخطاء

تركيب بيئة التطوير

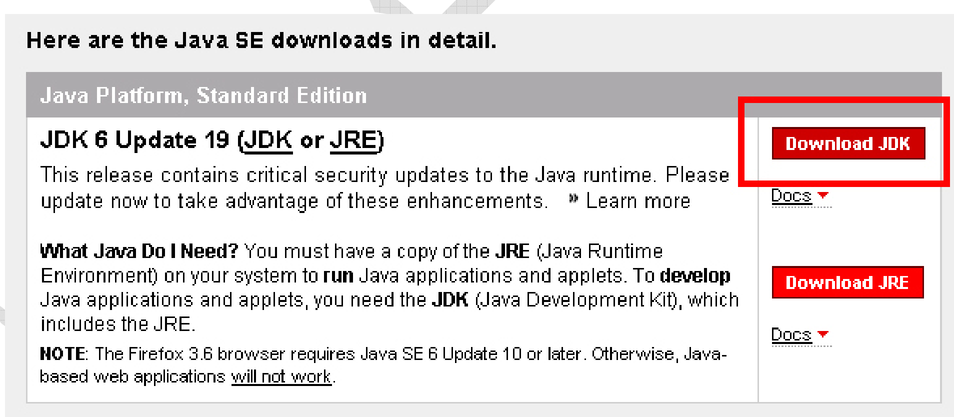
في عملية التركيب سأحاول قدر الإمكان عدم إغفال أي نقطة مهما كانت صغيرة و سأقوم بتوثيقها بالصور فكتابنا يدور حول البساطة و هي تبسيط كل شيء قد الإمكان لكي لا تقف خطوة صغيرة حجر عثرة في طريقنا لتعلم تطوير التطبيقات .

الخطوة الأولى (Java SE)

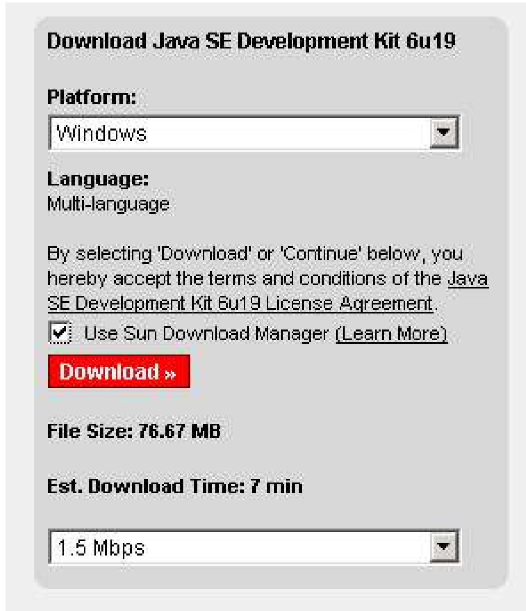
في البداية نحتاج إلى حزمة (Java SE) اختصارا إلى (Java Standard Edition) و تحديدا (JDK 6) و التي تحتوي على كل من (JDK)اختصارا إلى (Java Development Kit) و (JRE) اختصارا إلى (Java Runtime Environment) و لا يكتفي بوجود (JRE) فقط ولذا قم بتحميل (JDK) كاملة و التي تحتوي على الاثنين معا و التي يمكن تحميلها من موقع الانترنت:

http://java.sun.com/javase/downloads/index.jsp

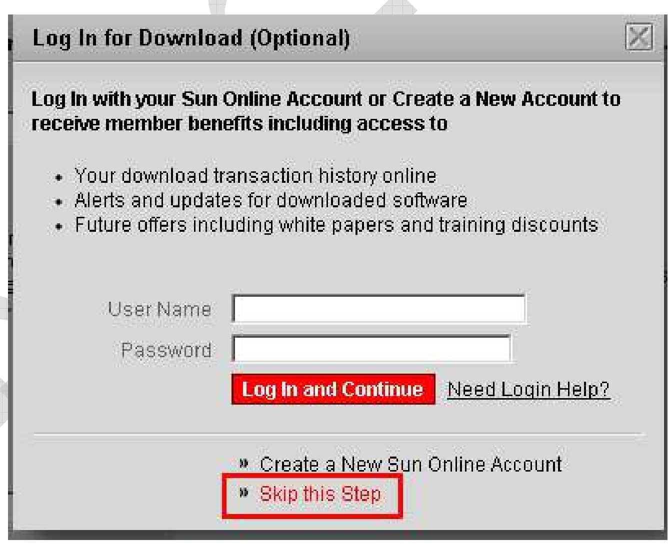
بعد فتح الصفحة ستظهر لك عدة خيارات اختر كما هو مبين ضمن الإطار:



في الشاشة التالية قم بتحديد نظام التشغيل الخاص بك كما في حالتنا هذه هو نظام تشغيل (Windows)

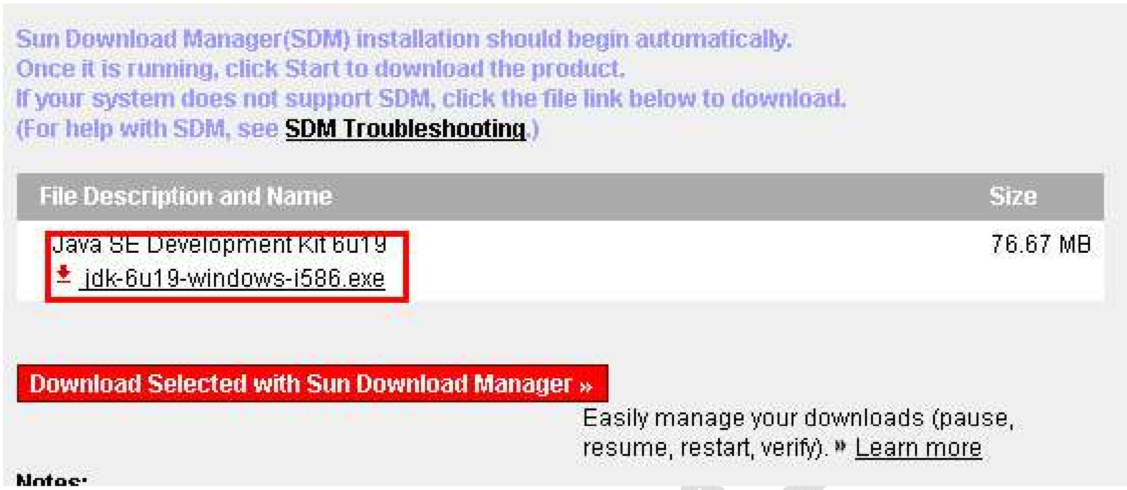


عند ظهور شاشة تطلب منك التسجيل قم بتجاهل ذلك عن طريق الضغط على (Skip) كما هو موضح في القوس التالي

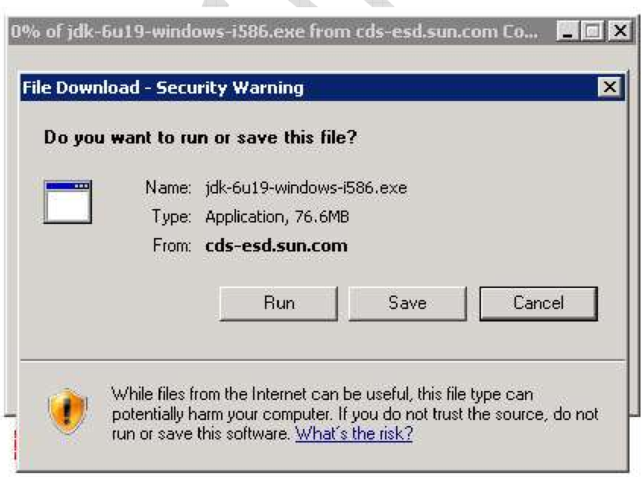


بعد ذلك سيخيرك الموقع في استخدام (Sun Download Manager) و الذي يساعدك في إكمال التحميل في حال حدث

انقطاع و كما يمكنك تحميل الملف مباشرة من الوصلة كما هو موضح في الصورة التالية



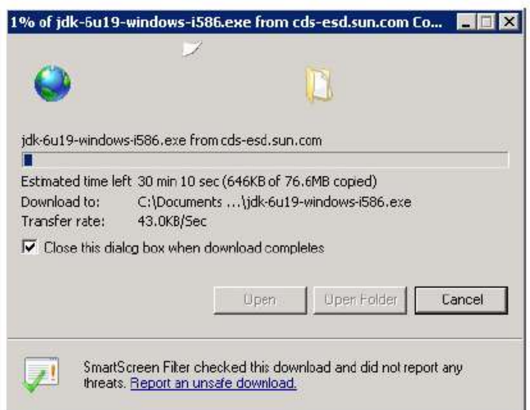
بعد ذلك ستظهر لنا شاشة تحميل الملف



اضغط على زر حفظ (Save) و عندها ستظهر لك الشاشة التالية و التي تمكنك من حفظ الملف في مكان محدد على جهازك



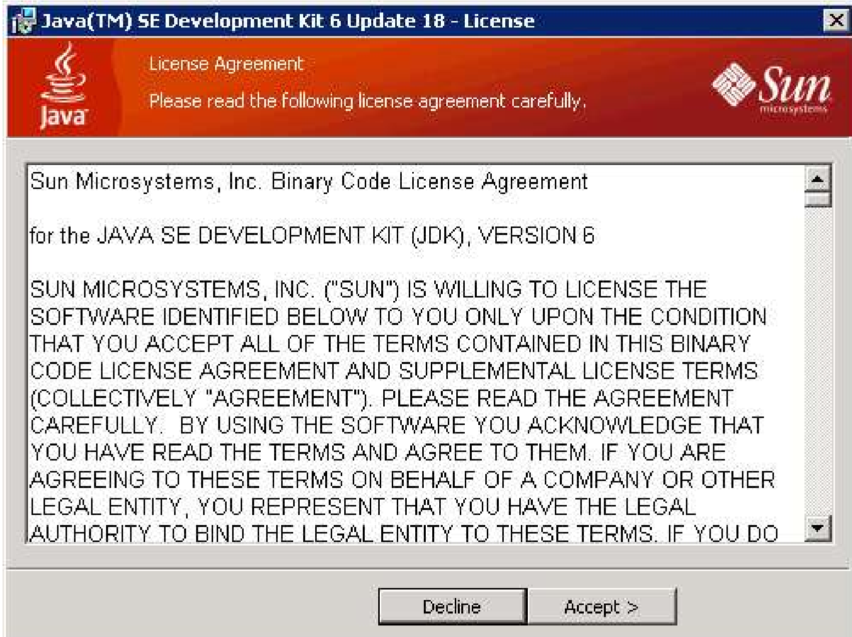
قم بحفظ الملف على سطح المكتب في الوقت الحال و لاحقا يمكنك الاحتفاظ بنسخة منه في مكان أمن و بعد حفظ الملف ستبدأ عملية التحميل و التي قد تستغرق بعد الوقت بناء على سرعة الانترنت لديك



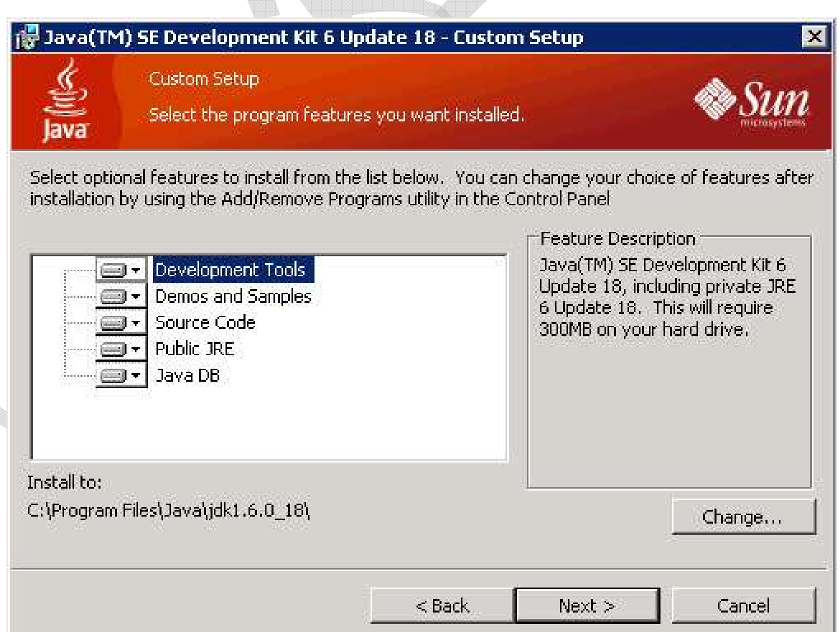
حجم الملف تقريبا (70) ميجا بايت ، ستجد أيقونته على سطح المكتب بالشكل التالي



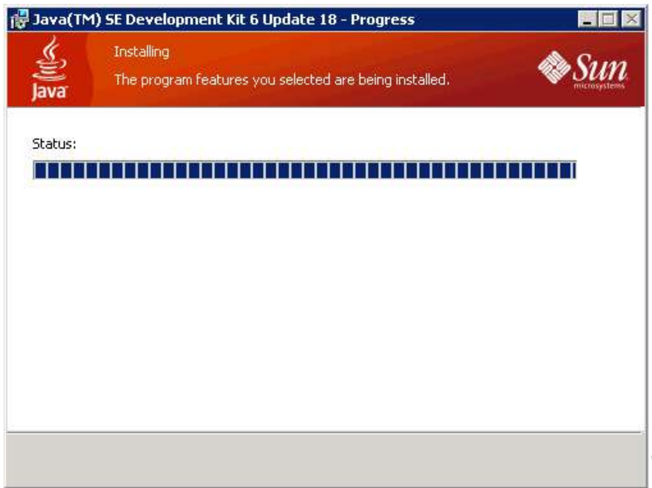
قم بالنقر مرتين على الملف لتشغيل برنامج الإعداد لتظهر لك شاشة شبيهة بهذه اضغط على قبول (Accept)



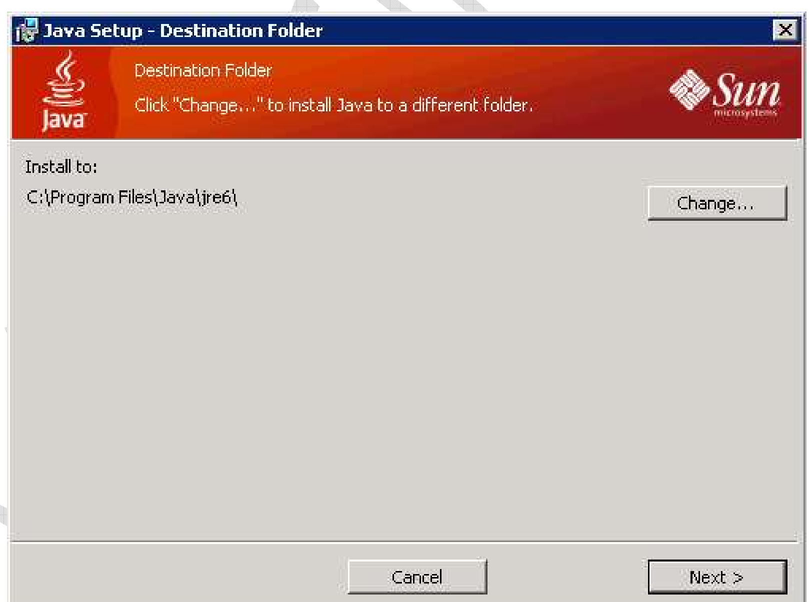
في الشاشة التالية سيسألك برنامج الإعداد عن المكونات التي تود تحميلها لا تقم بتغيير شيء و اضغط على زر التالي (Next)



سيبدأ البرنامج بفك نفسه على الجهاز تمهيدا لعملية الإعداد



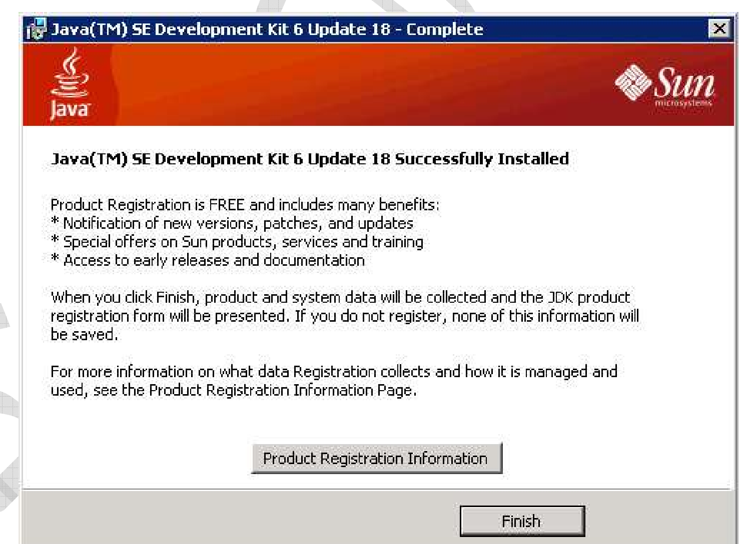
بعد دقائق قليلة ستظهر لك شاشة تخيرك فيها بتغيير المجلد الهدف للبرنامج لا تقم بتغيير شيء و اختر زر التالي (Next)



في هذه الخطوة يبدأ البرنامج في إعداد نفسه على الجهاز



بعد الانتهاء سيخبرك برنامج الإعداد بنجاحه في عملية الإعداد و عند الضغط على زر إنهاء (Finish) قد يطلب منك برنامج الإعداد ضرورة إعادة تشغيل الجهاز فقم بذلك و أعد تشغيل الجهاز



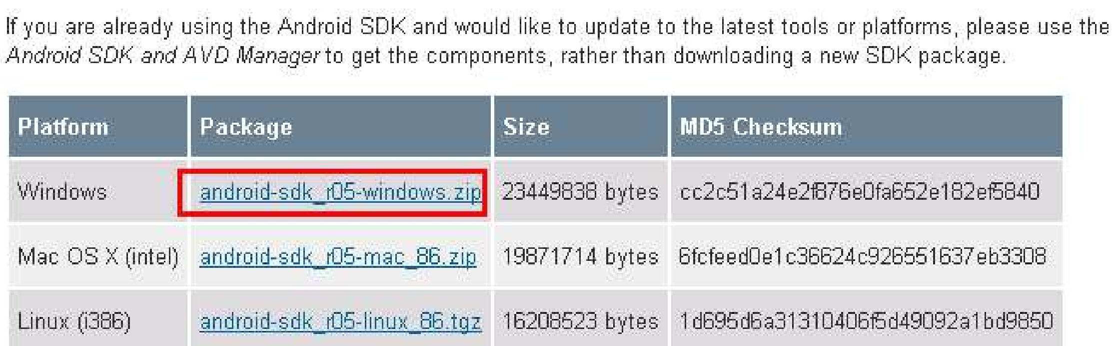
و بهذا نكون قد قمنا بالخطوة الأولى ألا و هي دعم النظام لبريمجات جافا

الخطوة الثانية (Android SDK)

تحميل حزمة مطوري التطبيقات الخاصة بأندرويد (Android SDK)

اذهب إلى العنوان التالي على الانترنت

http://developer.android.com/sdk/index.html



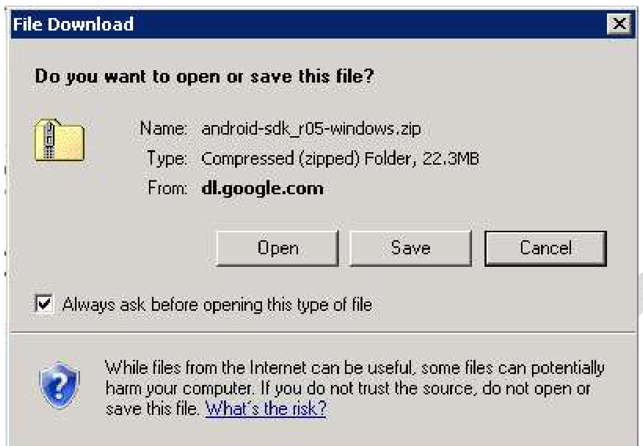
If you are already using the Android SDK and would like to update to the latest tools or platforms, please use the *Android SDK and AVD Manager* to get the components, rather than downloading a new SDK package.

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r05-windows.zip	23449838 bytes	cc2c51a24e2f876e0fa652e182ef5840
Mac OS X (intel)	android-sdk_r05-mac_86.zip	19871714 bytes	6fcfeed0e1c36624c926551637eb3308
Linux (i386)	android-sdk_r05-linux_86.tgz	16208523 bytes	1d695d6a31310406f5d49092a1bd9850

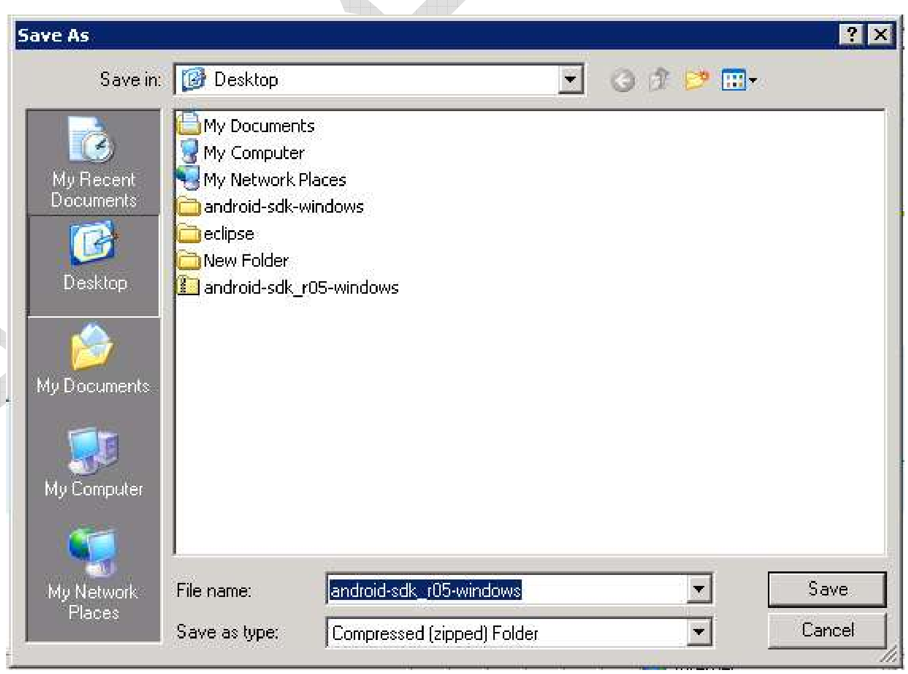
و هو الموقع المخصص لمطوري أندرويد من جوجل قم الآن بتحميل حزمة المطورين كما هو مبين أمامك بناء على نوع نظام التشغيل الخاص بك و في حالتنا هذه سنختار (Windows) ، و بعد ذلك سيقوم بتحويلنا إلى صفحة اتفاقية الاستخدام



انزل إلى أخر الصفحة و اختر الموافقة على اتفاقية المستخدم إن أردت - :) – و اضغط على زر تحميل (Download) و عندها ستظهر لنا شاشة تحميل البرنامج كما الشكل التالي و اضغط فيها على زر حفظ (Save)



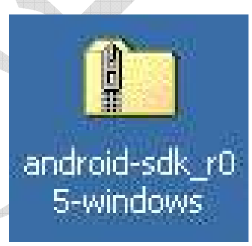
لتظهر لك بعد ذلك شاشة اختيار حفظ الملف كما الشكل التالي



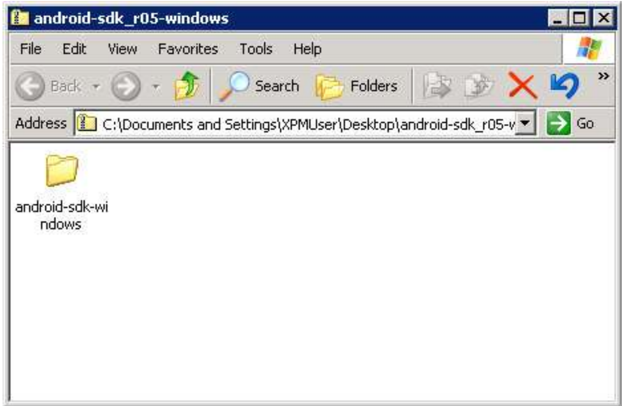
قم بتحديد وجهة الحفظ و لتكن سطح المكتب ثم انقر على زر حفظ (Save) و عندها ستبدأ عملية تحميل البرنامج على جهازك



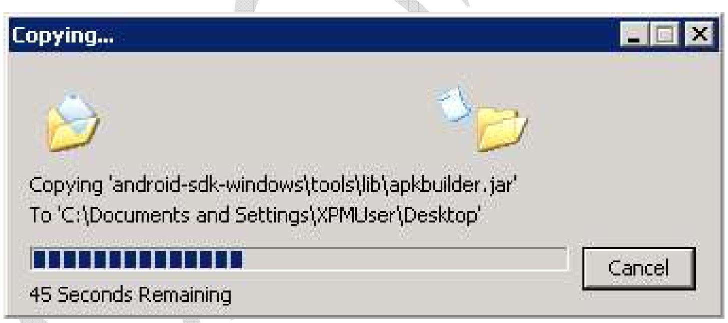
حجم البرنامج تقريبا (22) ميجابايت و بعد الانتهاء سيظهر لك ملف مضغوط على سطح المكتب كما الشكل التالي



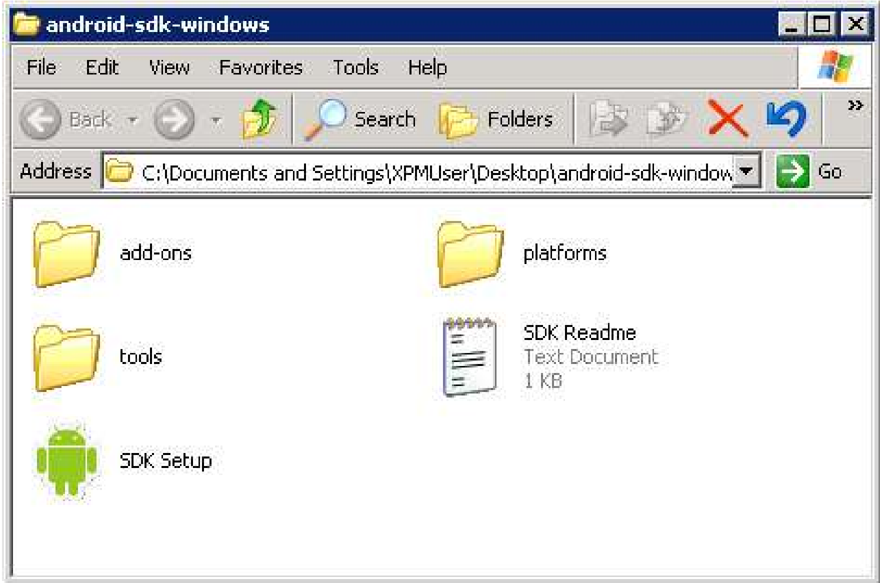
يمكنك فك ضغط الملف باستخدام برامج فك الضغط مثل (WinZip) إن كنت تملكه و في حال لم تكن تملكه فمجرد الضغط عليه فإن نظام التشغيل (Windows) يدعم هذه الخاصية و سيفتح مجلد به محتويات الملف المضغوط كما الشكل التالي



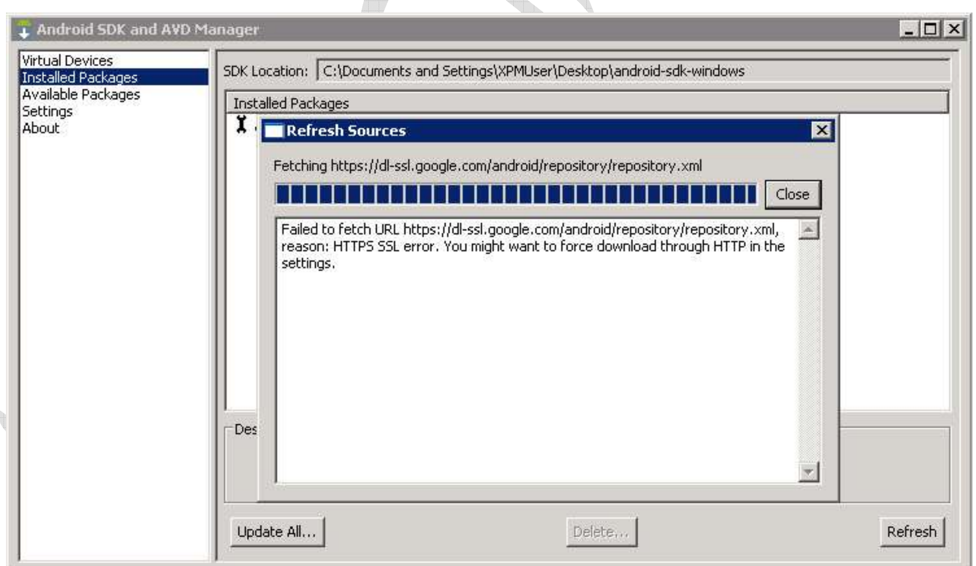
الآن قم بسحب المجلد إلى سطح المكتب عن طريق السحب و الإفلات بزر الفأرة أو قم بالضغط بزر الفأرة الأيمن عليه و اختيار نسخ (Copy) ثم أذهب إلى سطح المكتب و اختر من قائمة زر الفأرة الأيمن لصق (Paste) و عندها ستبدأ عملية فك الضغط كما في الشاشة التالية



بعد الانتهاء من عملية فك الضغط ستجد مجلد جديد على سطح المكتب باسم الملف (android-sdk-windows) قم بالضغط المضاعف على المجلد لفتحه ليظهر لك محتوياته



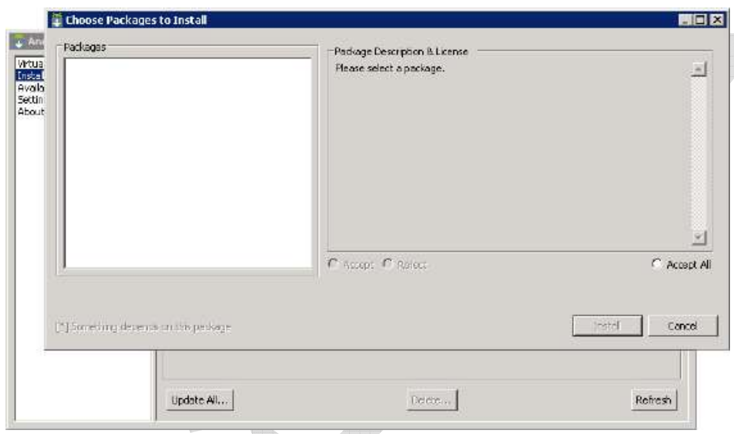
قم بتشغيل الملف التنفيذي (SDK Setup) لتبدأ في عملية تحميل أخر ملفات الإعداد من موقع أندرويد و ستظهر لك الشاشة التالية



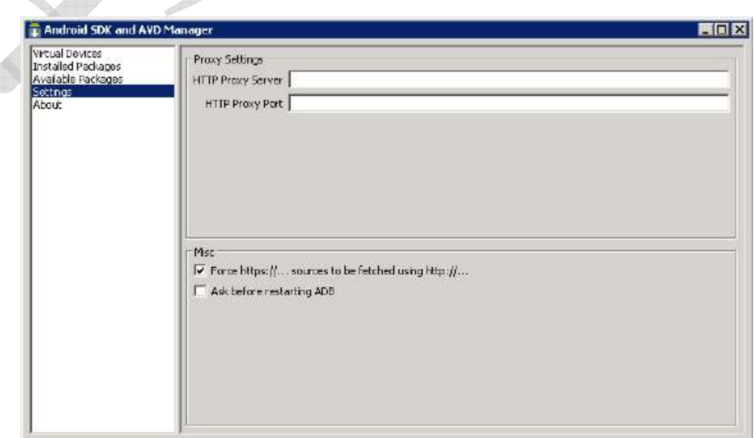
كما ترى في حالتنا هذه تظهر الشاشة وجود عملية خطاء أثناء عملية التحميل و ذلك لعدم تمكن برنامج الإعداد من التخاطب عبر بروتوكول (https) و قد لا تظهر لك الشاشة في حال لم يكن على

جهازك أي إعدادات أمان مخصصة و لكن الكثيرين ممن حاولوا في عملية التحميل واجهوا هذه المشكلة و لذا سنقوم بشرح طريقة حلها و من ثم ننتقل للخطوة التي تليها و التي في حال لم تواجهك أي مشاكل منذ البداية من المفروض أن تظهر لك

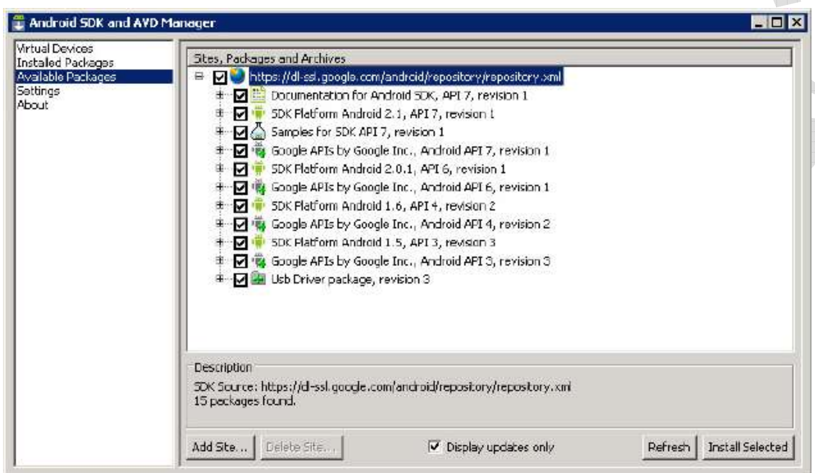
الآن اضغط على زر إغلاق (Close) لتظهر لك شاشة أخرى مثل هذه



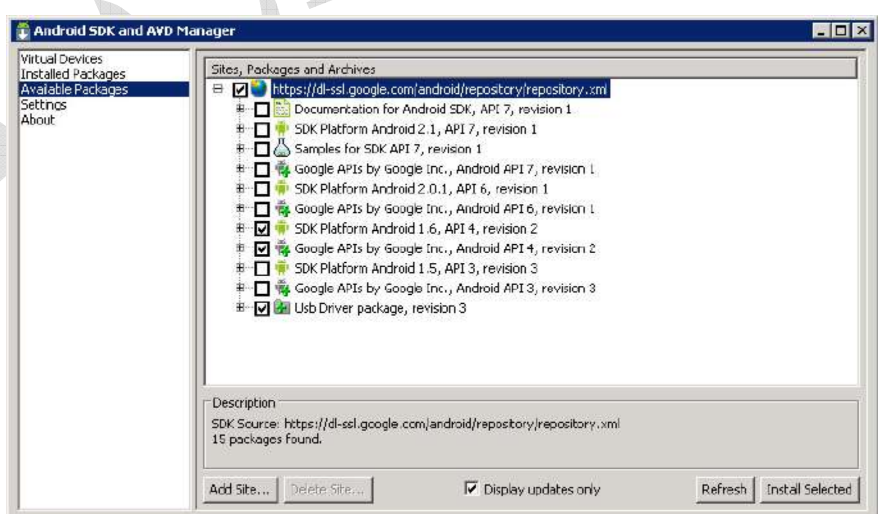
قم بالضغط علة زر إلغاء (Cancel) لتظهر لك الشاشة الرئيسية لبرنامج الإعداد كما الشكل التالي و عندها اذهب إلى الاختيار إعدادات (Settings) لتظهر لك شاشة شبيه بالتالية



الآن قم باختيار زر الخيار (Force Https) ليتغاضى عن العمل ضمن بروتوكول (Https) و في حال كنت تستخدم إعدادات بروكسي خاصة على جهازك قم بكتابتها في المربعات الفارغة في الأعلى ثم و من نفس الشاشة اذهب الخيار الذي يعلوه (Available Packages) لتظهر لك شاشة شبيهة بالتالية

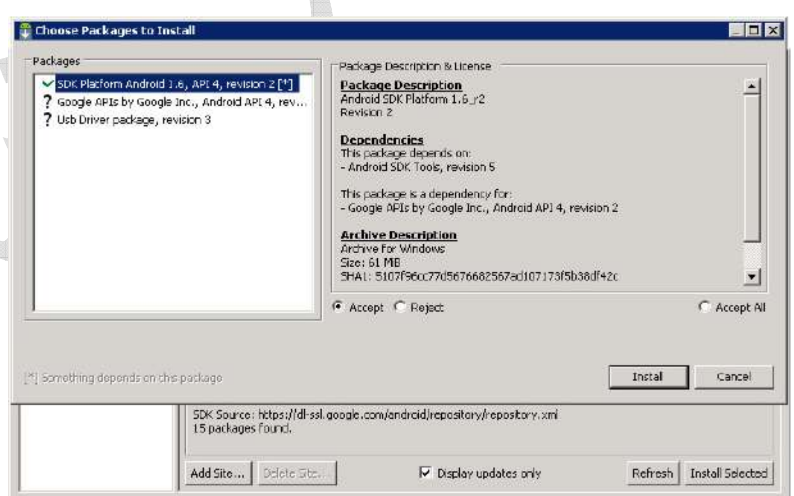


و عند فرد الشجرة ستظهر لك كل الأنظمة و المتوفرة حول أندرويد و بالنسبة لحالتنا يمكنك الإبقاء عليها جميعا معلمة أو تقوم فقط بتحديد (1.6) و الذي يوف ستكون حوله أمثلتنا في هذا الكتاب

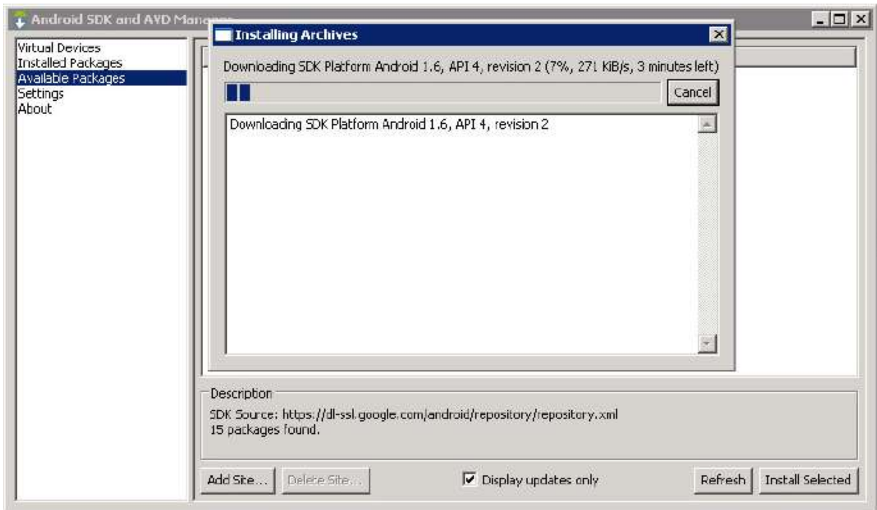


قد يتساءل البعض لماذا اخترنا (1.6) و لم نختر أحدث نظام (2.1)؟ و السبب ببساطة هو أن قرابة (50%) من الأجهزة المتوفرة في الأسواق تعمل بنظام (1.6). كما أن النسخ الأحدث لم تتوفر في أسواقنا العربية بعد و الأهم من ذلك فإن أندرويد يدعم بشكل تلقائي في نظم تشغيله الأحدث التطبيقات التي صممت على نظام سابق. كما أن أكثر التطبيقات المتوفرة في الأسواق موجهة لهذا النظام و في حال كنت مثلي و تملك جهاز على نظام (1.5) و هو المتوفر في الأسواق العربية فإن تطبيقك سيعمل عليه أيضا بدون أي مشاكل و بالنهاية لك الخيار في تحميلها جميعا أو الاكتفاء بهذا حيث أن كل عنصر من هذه تحتاج إلى وقت طويل في عملية التحميل و يمكنك في أي وقت العودة و تحميلها بإتباع نفس الخطوات السابقة.

الآن و بعد اختيار العناصر هدف التحميل قم بالضغط على زر (Install Selected) و الموجود في أسفل يمين الشاشة لتظهر لك شاشة جديدة شبيهة بالتالية و التي تحتوي اتفاقية الاستخدام



قم بالموافقة عبر اختيار (Accept All) للموافقة على الجميع مرة واحدة و من ثم اضغط على زر (Install) لتظهر لك شاشة شبيه بالتالية



بعد الانتهاء من عملية التحميل و التي قد تأخذ بعض الوقت قم بإغلاق النافذة و بهذا نكون قد انتهينا من تحميل حزمة التطوير و لننتقل إلى الخطوة الثالثة..

الخطوة الثالثة (Eclipse)

بيئة التطوير (Eclipse) هي بيئة تطوير متعددة الاستخدام كانت ملك شركة (IBM) و من ثم قامت منذ بضع سنوات بتوفيرها بشكل مجاني و مفتوح المصدر و لذا تجد أن أشهر المبرمجين يعملون عليها لتطوير تطبيقاتهم إذ تدعم عدة منصات عبر تركيب الإضافات الخاصة بكل منصة عن طريق نفس بيئة التطوير.

في البداية قم بزيارة موقع البرنامج على الوصلة التالية:

http://www.Eclipsee.org/downloads

في الصفحة الظاهرة أمامك ستجد عدة إصدارات من التطبيق لا تشعر بالارتباك و اذهب إلى أخر الصفحة لنقوم بتحميل النسخة الأساسية فقط و هي كل ما نحتاج (Eclipse Classic 3.5.2)

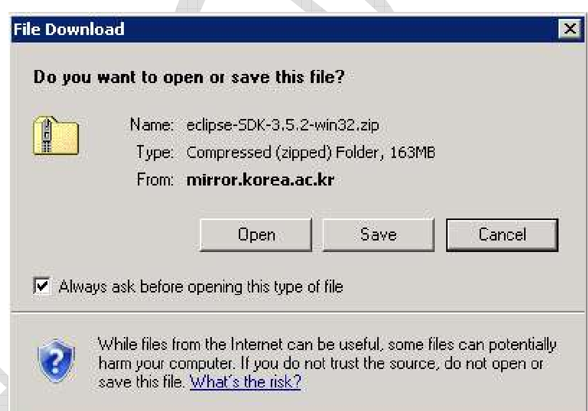


قم بالضغط عليها لتأخذك إلى الصفحة التالية

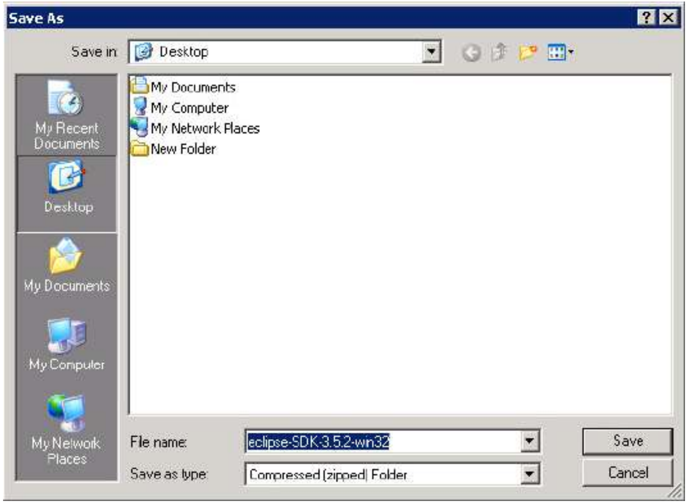


عن طريق هذه الصفحة يمكن اختيار طريقة التحميل إما عن طريق بروتوكول (Torrent) أو بشكل مباشر عن طريق أحد الخوادم التي توفر التطبيق ، لك الخيار في ذلك و لكني أفضل دائما استخدام الوصلات المباشرة لعدم حاجتها إلى أي تطبيقات أخرى كما هو موضح في المربع في الصورة.

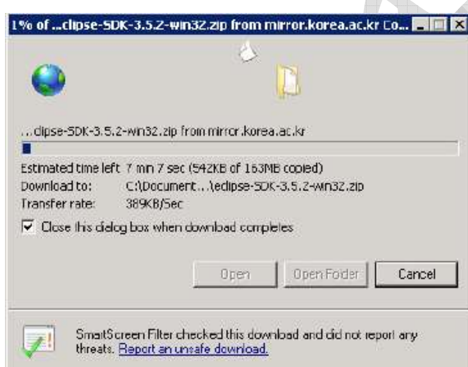
بعد اختياره ستظهر لك شاشة بدء التحميل كما في الصورة التالية



قم بالضغط على زر حفظ (Save) و من ثم في الشاشة التالية قم بتحديد هدف الحفظ على جهازك و ليكن سطح المكتب



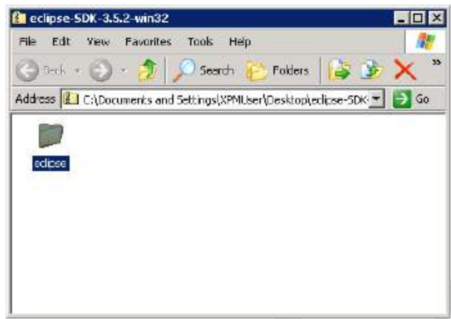
بعد الضغط على زر الحفظ (Save) ستظهر لك شاشة بدء التحميل كما في الصورة التالية



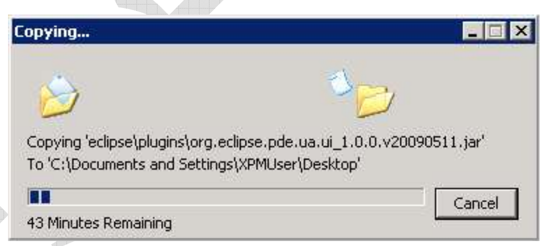
قد تأخذ منك العملية بعض الوقت بناء على سرعة الانترنت لديهم و عند انتهائها ستجد ملف مضغوط باسم (Eclipsee-SDK-3.5.2-win32) على سطح المكتب كما في الصورة التالية



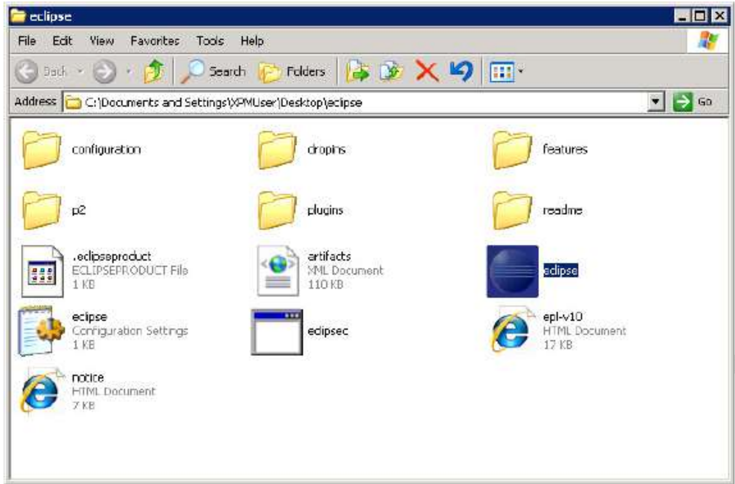
حجم البرنامج قرابة (163) ميجابايت و في حال كنت تملك برنامج لفك الضغط مثل (Win Zip) فيمكنك فك البرنامج على سطح المكتب مباشرة أو قم بالضغط المضاعف عليه ليقوم نظام التشغيل (Windows) بفتحه كما في الشكل التالي:



قم الآن بسحب المجلد إلى سطح المكتب عبر خاصية السحب و الإفلات بالفأرة أو قم بالضغط بزر الفأرة الأيمن و من ثم اختيار خيار نسخ (Copy) ومن ثم اذهب إلى سطح المكتب و قم بزر الفأرة الأيمن باختيار خيار (Paste) لتبدأ عملية فك ضغط الملف:



بعد هذه الخطوة ستجد مجلد على سطح المكتب لديك باسم (Eclipse) قم بالضغط المضاعف عليه ليفتح لديك محتوياته

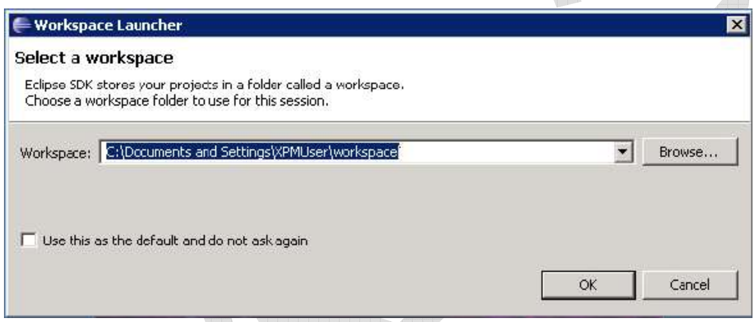


وبهذا نكون قد انتهينا من عملية تنصيب بيئة التطوير إذ أنـها من نوعية التطبيقات المحمولة و التي لا تحتاج إلى عملية إعداد على الجهاز و تعمل مباشرة عن طريق تشغيل البرنـامج التنفيذي مباشرة بهذا نكون قد انتهينا من الخطوة الثالثة و لننتقل للخطوة التي تليها

الخطوة الرابعة (ADT)

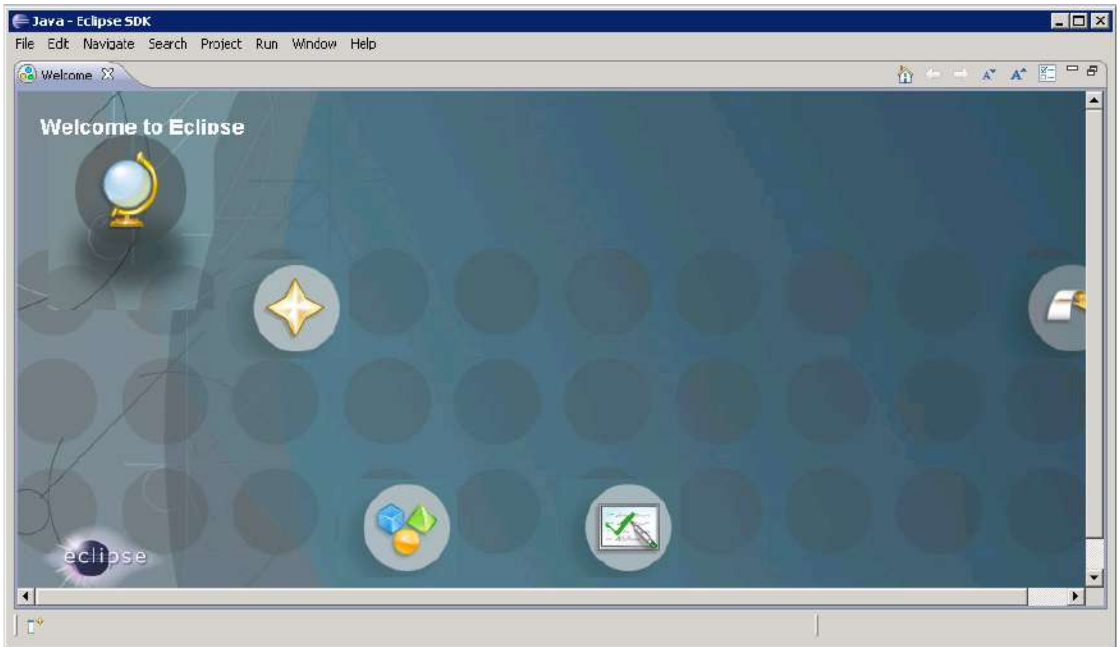
تحميل إضافة (ADT) لدعم التطوير الموجهة لتطبيقات أندرويد على (Eclipse)

بعد انتهاء الخطوة السابقة قم بتشغيل برنامج (Eclipse) لتظهر لك الشاشة التالية لأول مرة

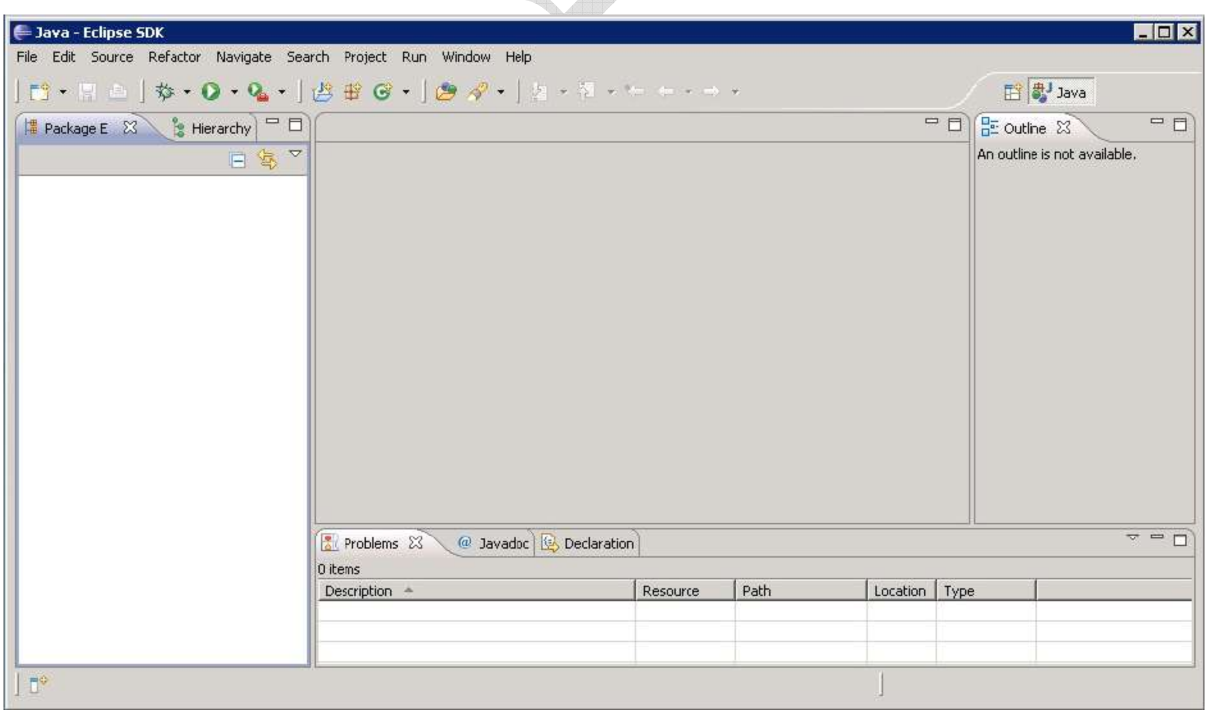


وهذه الشاشة التي تحدد فيها مساحة العمل الخاصة بك و التي يتم تخزين ملفات مشاريعك فيها و يمكنك تغيير مكانها لحقا أو اختيار خيار (Use this as default and do not ask again) و التي تحفظ اختيارك لكي لا تظهر لك في كل مرة تقوم فيها بتشغيل بيئة التطوير

اضغط الآن على زر موافق (OK) لتظهر لك الشاشة الأولى من برنامج التطوير والشبيهة بالتالية



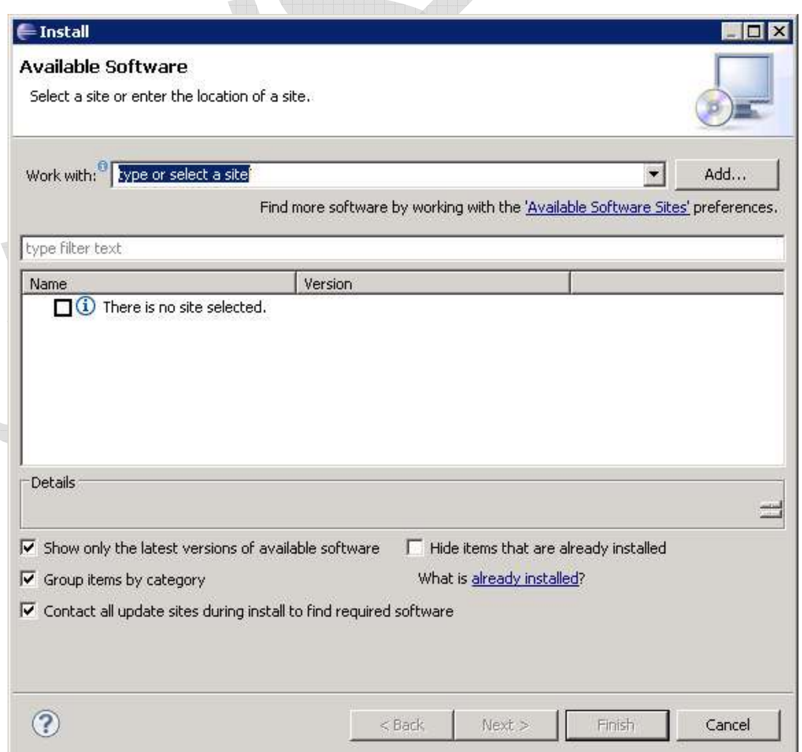
في أعلى يسار الشاشة تجد لسان تبويب الترحيب (welcome) قم بضغط زر (X) لإغلاقه لتظهر لك شاشة بيئة التطوير كما الشكل التالي



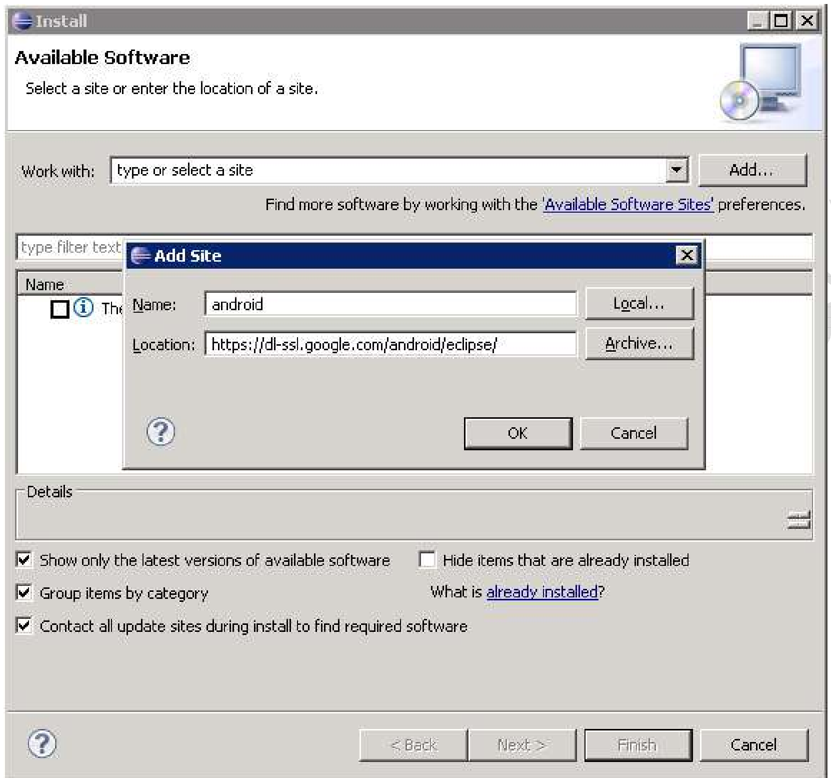
الآن لنقوم بتحميل إضافة (ADT) قم بالذهاب القائمة (Help) و منها قم باختيار خيار (Install New Software) كما هو موضح بالصورة التالية



بعد اختيارك لهذا الخيار سوف تظهر لك شاشة شبيهة بالشاشة التالية



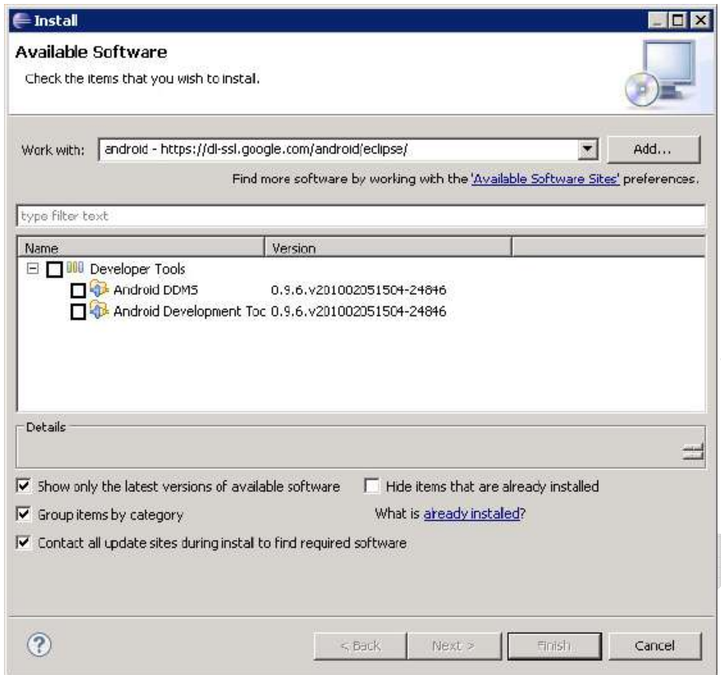
قم بالضغط علة زر (Add) لإضافة عنوان التطبيق لتظهر لك شاشة شبيهة بالشاشة التالية



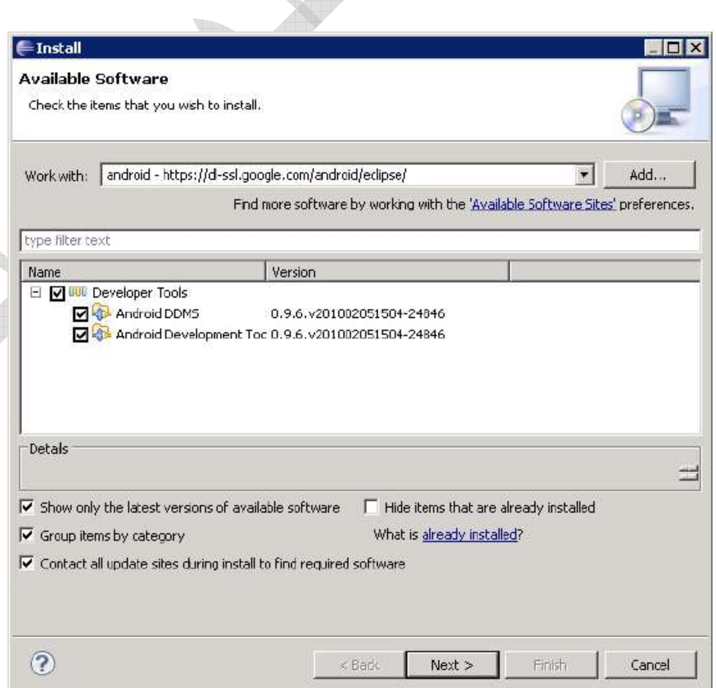
قم بكتابة (Android) في مربع الاسم (Name) و قم بكتابة العنوان التالي في مربع (Location)

/https://dl-ssl.google.com/android/Eclipsee

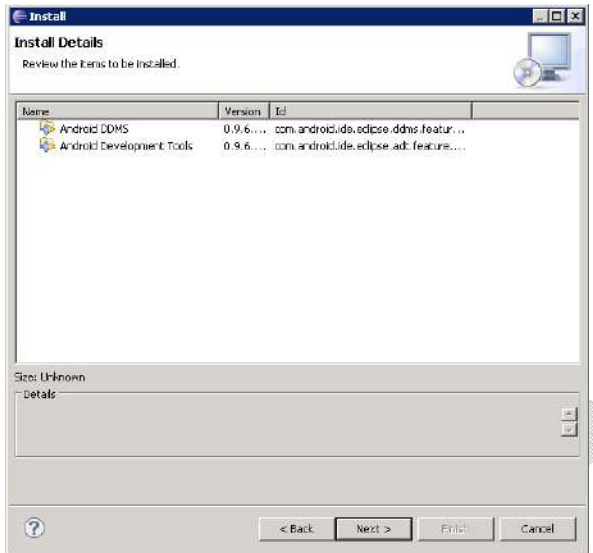
ومن ثم اضغط على زر موافق (OK) و بعد دقائق معدودة سيقوم البرنامج بالاتصال بالانترنت للبحث عن الملفات الخاصة بالتطبيق لتظهر لك شاشة شبيهة بالتالية



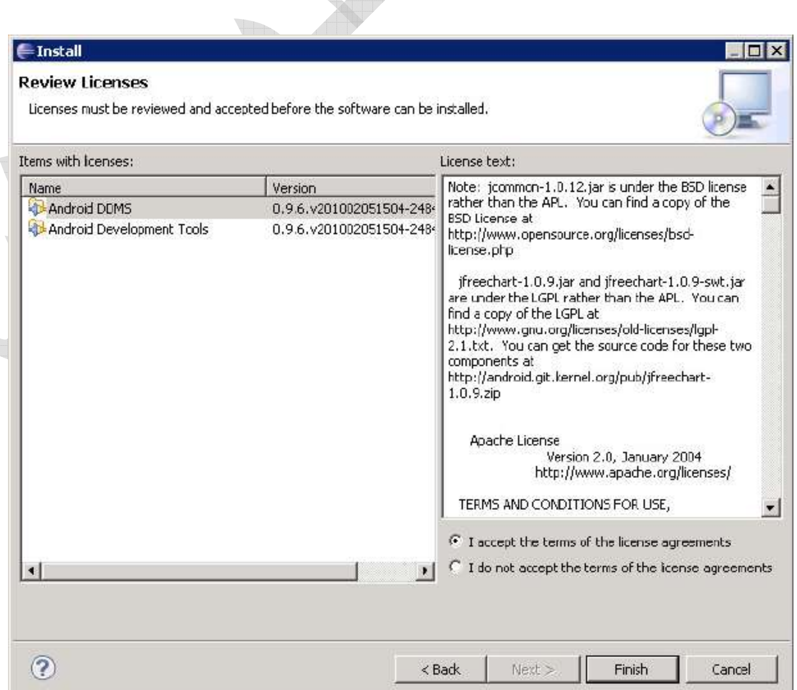
في حال لم تظهر لك هذه الشاشة قم بإزالة حرف (S) من عنوان الوصلة لتصبح من (https) إلى (http) ثم باقي الوصلة كما هي و عندها بالتأكيد ستظهر لك شاشة شبيهة بالشاشة السابقة



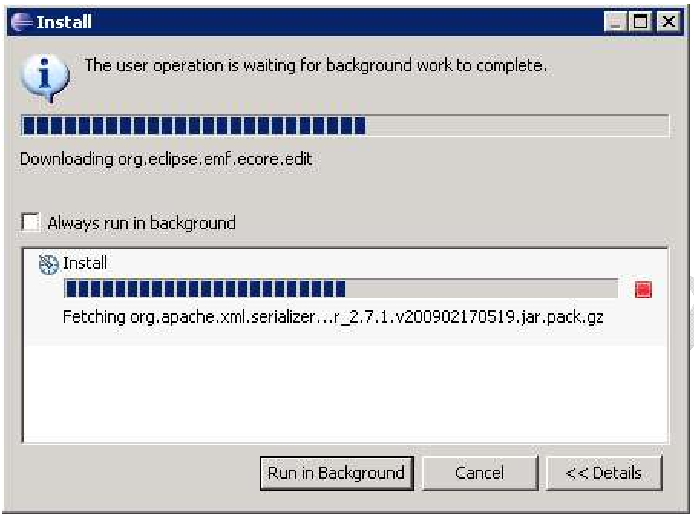
قم بتعليم كل العناصر و من ثم اضغط على زر التالي (Next) لتظهر لك الشاشة التالية



ثم اضغط على زر التالي لتظهر شاشة اتفاقية المستخدم كما الصورة التالية



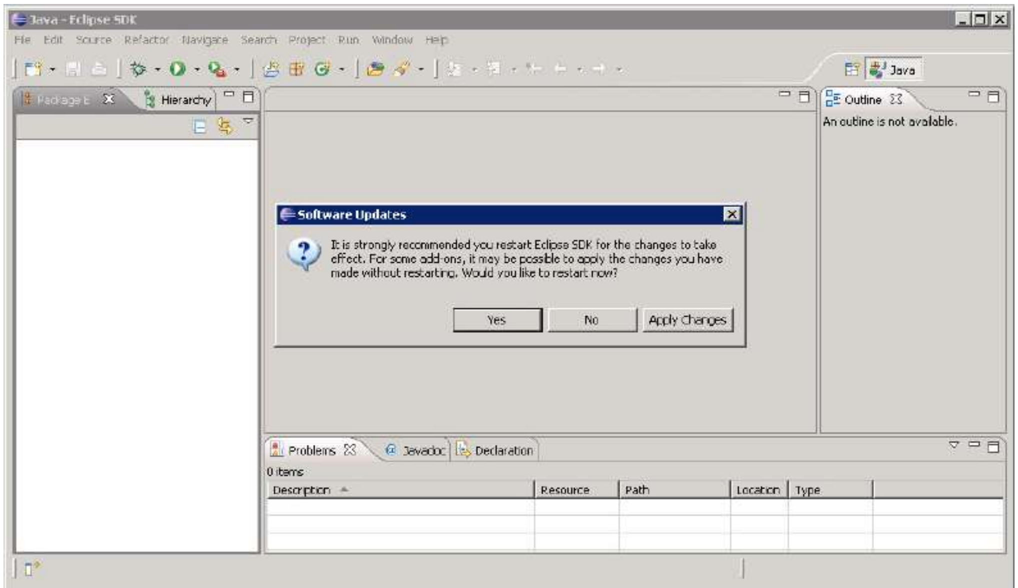
قم بالموافقة على اتفاقية المستخدم عبر اختيار خيار (I accept) و من ثم اختر زر إنـهاء (Finish) لتظهر لك شاشة بدء التحميل كما الصورة التالية



و بعد انتهائه من التحميل عبر الانترنت ستظهر لك شاشة تحذير بأن هناك المزيد من التطبيقات التي تود التنصيب على جهازك قم بالضغط على زر موافق



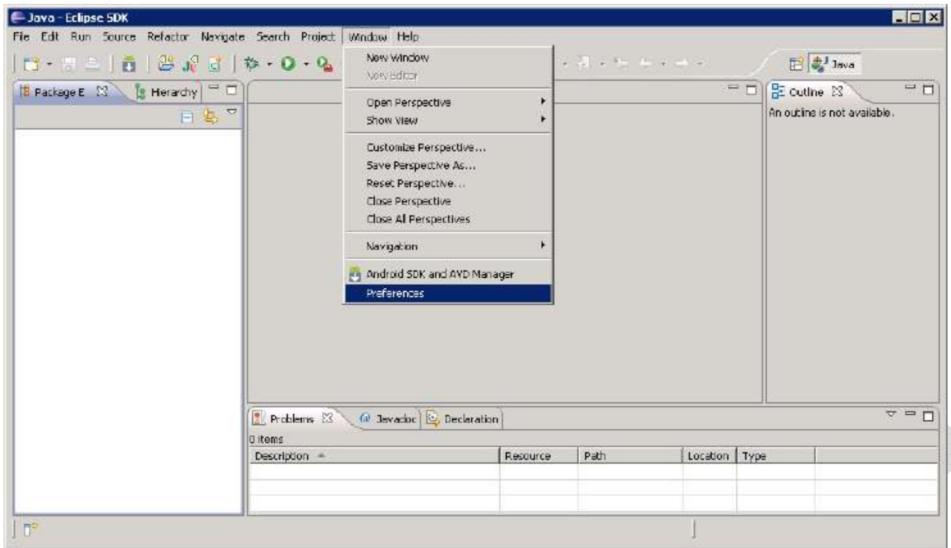
ثم بعد الانتهاء من عملية التنصيب ستظهر لك شاشة أخرى تخبرك بأنك بحاجة إلى إعادة تشغيل تطبيق بيئة التطوير (Eclipse) لتصبح الإعدادات الجديدة نافذة فاضغط على زر موافق



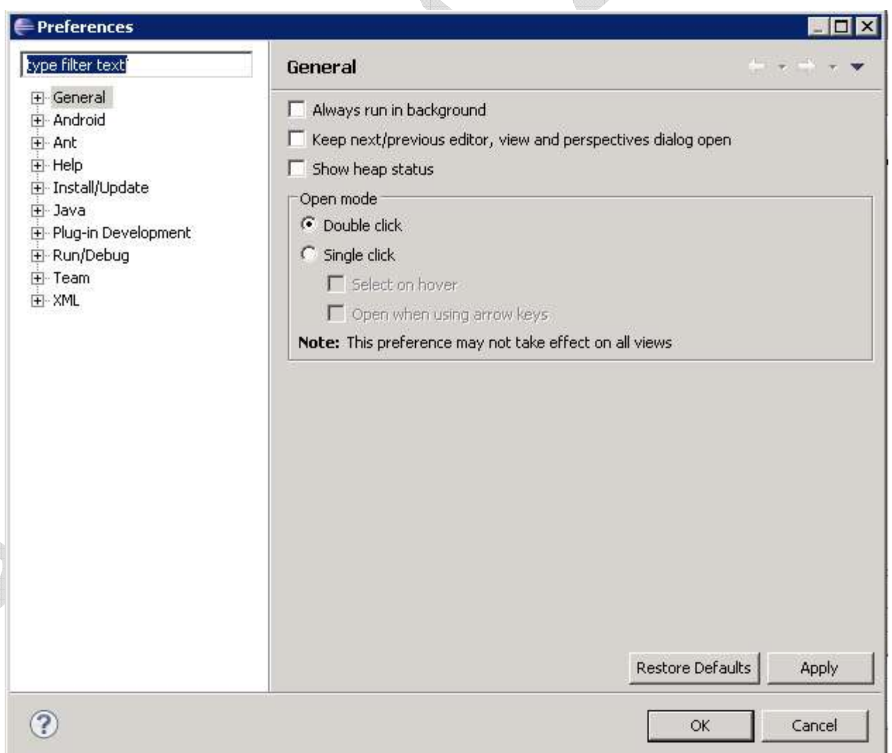
ستقوم بيئة التطوير بالإغلاق بشكل ألي و من ثم ستعاود التشغيل مرة أخرى و بهذا تصبح بيئة التطوير لديك إمكانية البرمجة الموجهة لاندرويد.

بهذا نكون قد انتهينا من الخطوة الرابعة و لننتقل للخطوة الخامسة و الأخيرة و بعدها نبدأ في البرمجة.

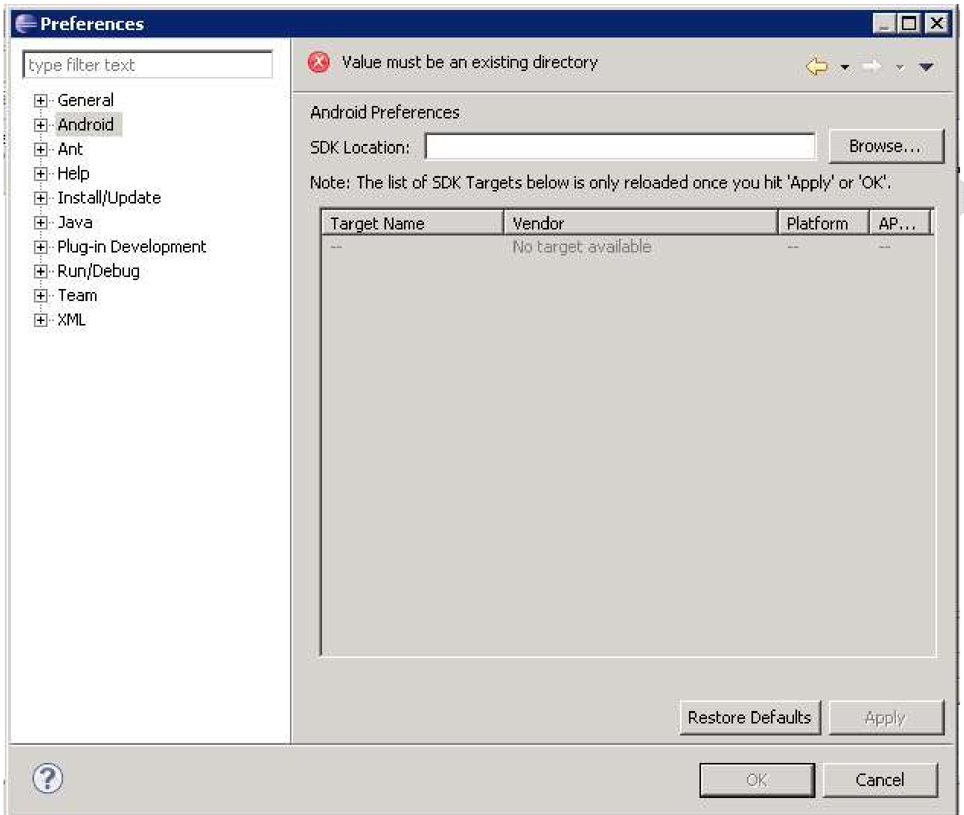
من القائمة window قم باختيار خيار (Preferences) كما هو موضح بالصورة التالية:



لتظهر لك شاشة شبيهة بالتالية:



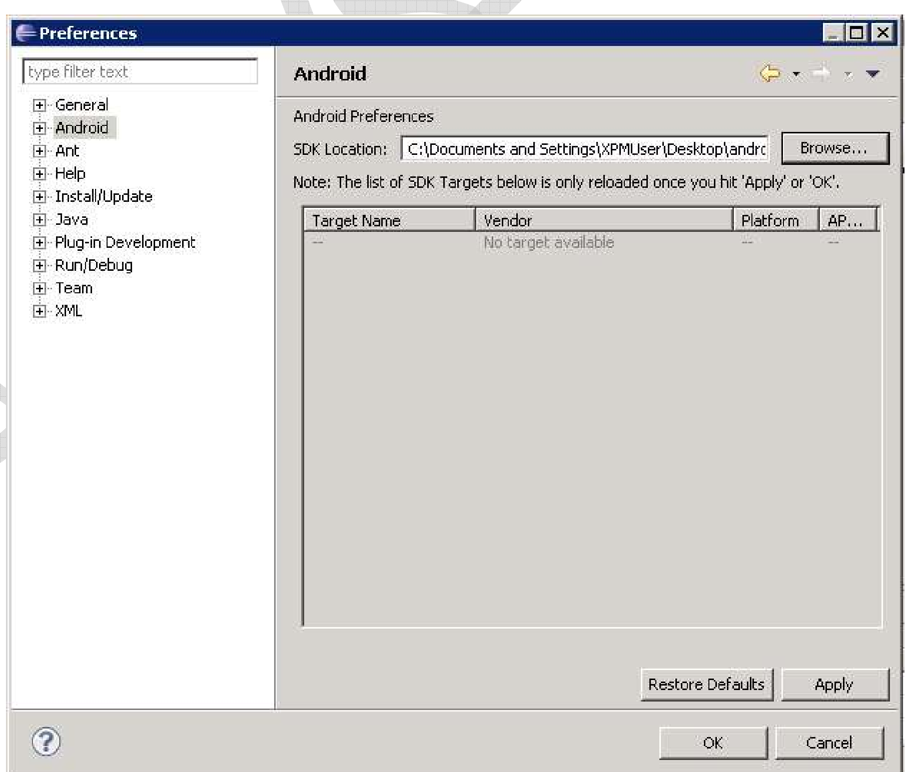
من أعلى الشاشة في الجهة اليسرى قم باختيار (Android) و قد تظهر لك شاشة خطاء و في حال ظهورها اضغط على موافق وتجاهلها لتصبح لك الشاشة بالشكل التالي:



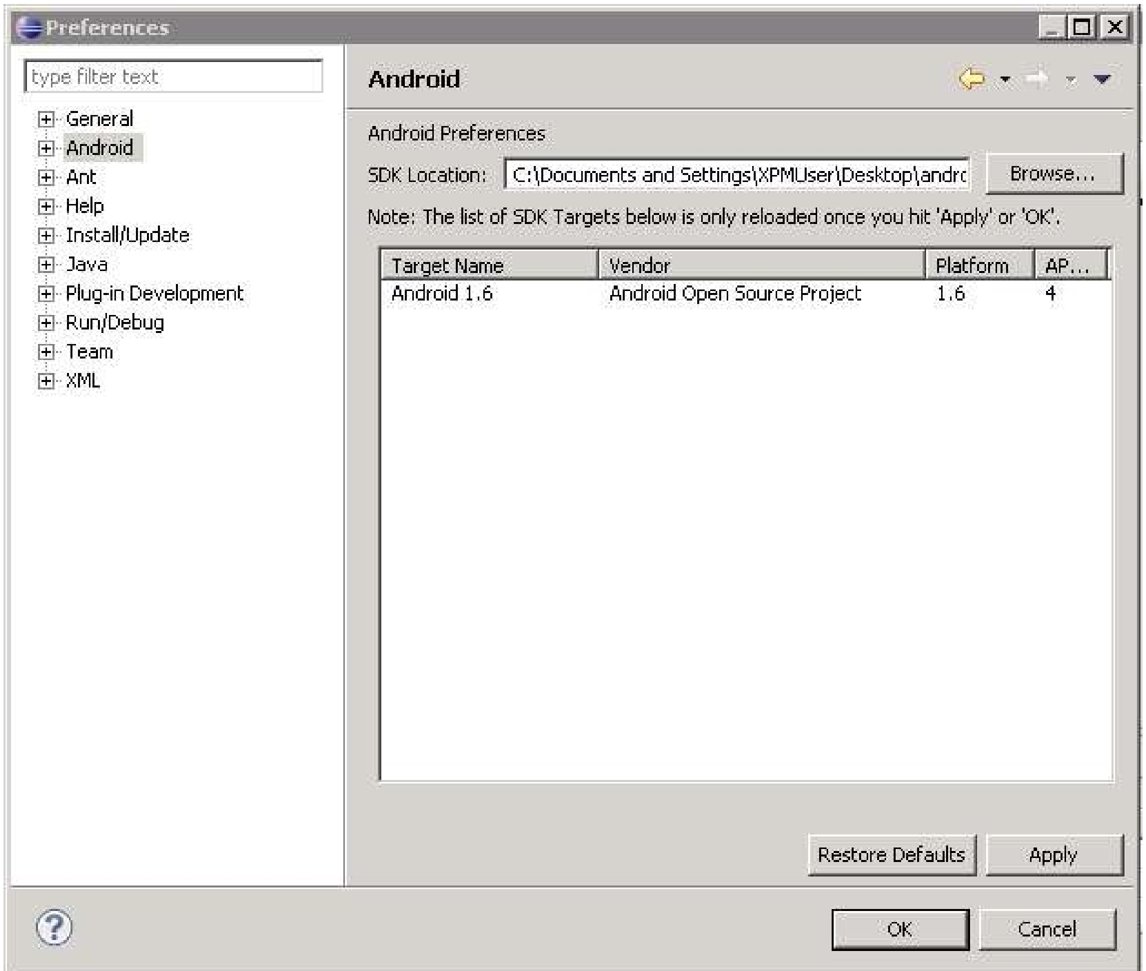
الآن من زر استعراض (Browse) قم بالبحث عن مكان ملفات حزمة التطوير و التي قمنا سابقا في تحميلها على جهازنا و التي كانت على سطح المكتب و نختار المجلد الرئيسي كما الصورة التالية:



ثم نضغط على زر موافق (OK) لتصبح الشاشة لدينا كما في الشكل التالي:



الآن قم بالضغط على زر (Apply) و الموجود في أسفل يمين الشاشة لتطبيق الإعدادات الجديدة لتظهر لك شاشة شبيهة بالشاشة التالية:



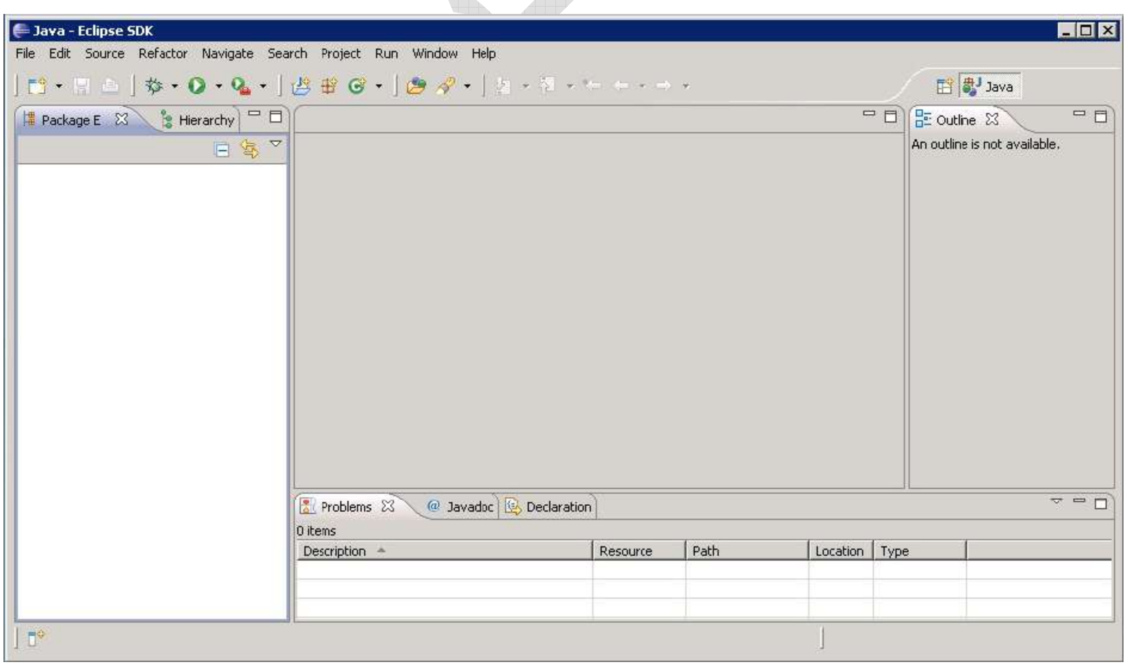
و التي تقوم بعرض المنصات التي قمنا بتحميلها سابقا أثناء عملية تنصيب حزمة التطوير .

قم بالضغط على زر موافق لإغلاق النافذة و الآن انتهينا من عملية إعداد بيئة التطوير و كل توابعها بشكل كامل و لننتقل للخطوة التالية للتعرف أكثر على بيئة التطوير و كيفية التعامل معها.

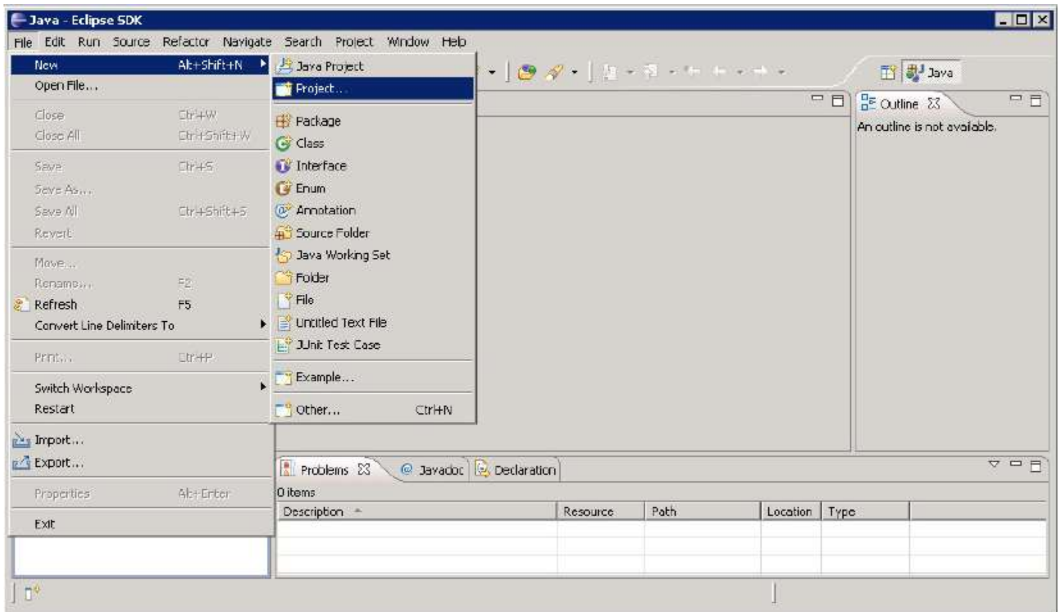
برنامجك الأول

لربما و قبل الإبحار في البرمجة دعنا نمتع أعيننا بالتطبيق الأول و نشاهد مدى سهولة العمل حيث أننا سنقوم ببناء تطبيقان الأول و الذي تعارف عليه لدى كل لغات البرمجة باسم (Hell World) (أهلا بالعالم) و الذي يعني لحظة ولادة التطبيق الأولى و التي خرج بها إلى العالم و يبدأهم التحية.

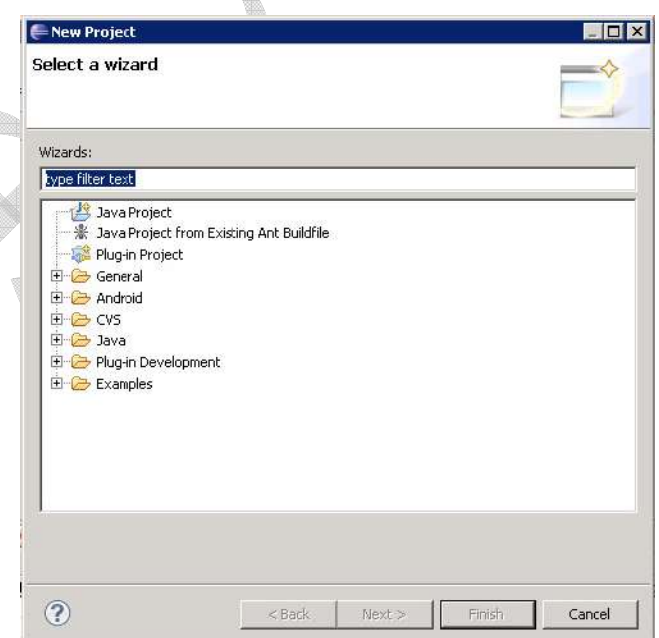
إلا أني أرى أن (أندرويد ببساطة) أسهل و لذا سنبني تطبيق أندرويد ببساطة و بكل بساطة و بدون سطر برمجي واحد و بعد أن نقوم ببناء التطبيق و التعرف على خطوات بدء مشروع جديد سنقوم بإعداد محاكي جهاز أندرويد لتستطيع تشغيل برنامجك عليه.
إن كنت قد قمت بإغلاق بيئة التطوير (Eclipse) فقم بإعادة تشغيلها الآن لتظهر لنا الشاشة الرئيسية



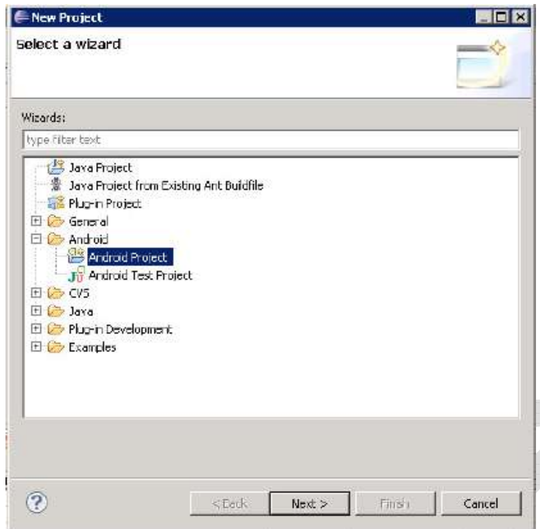
من القائمة العليا ملف (File) قم باختيار (New) ثم (New Project) أي مشروع جديد كما هو موضح بالصورة التالية



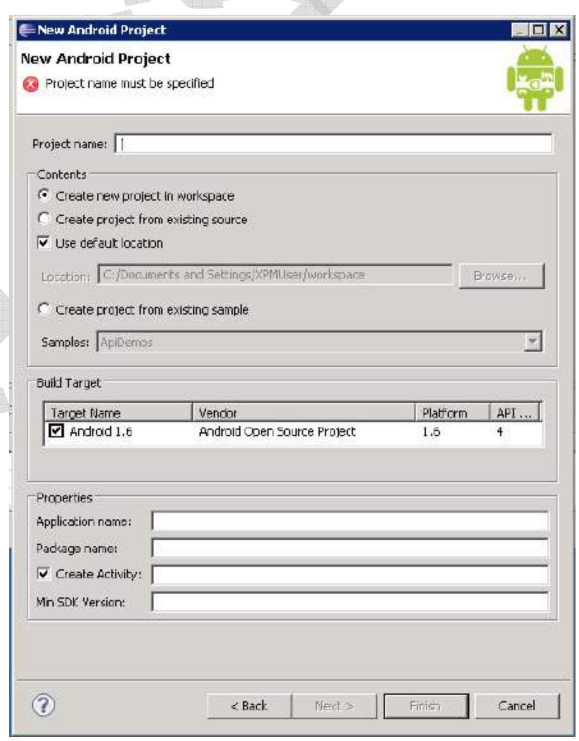
عن الضغط على زر مشروع جديد ستظهر لك شاشة توصيف المشروع و هي شبيهة بالشكل التالي



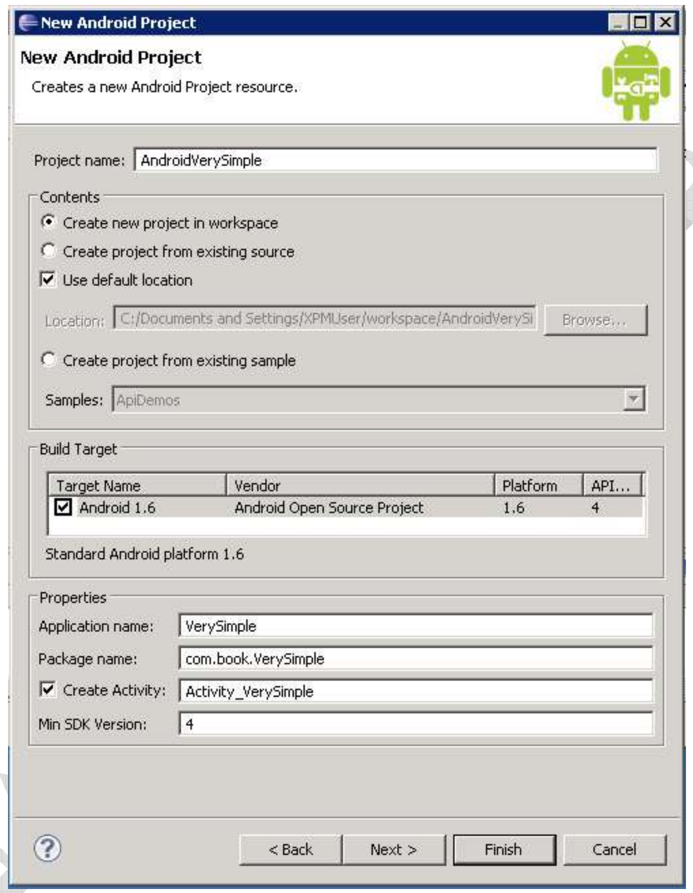
اختر منها مشروع أندرويد كما الصورة التالية



ثم اضغط على زر التالية (Next) لتظهر لك الشاشة التالية

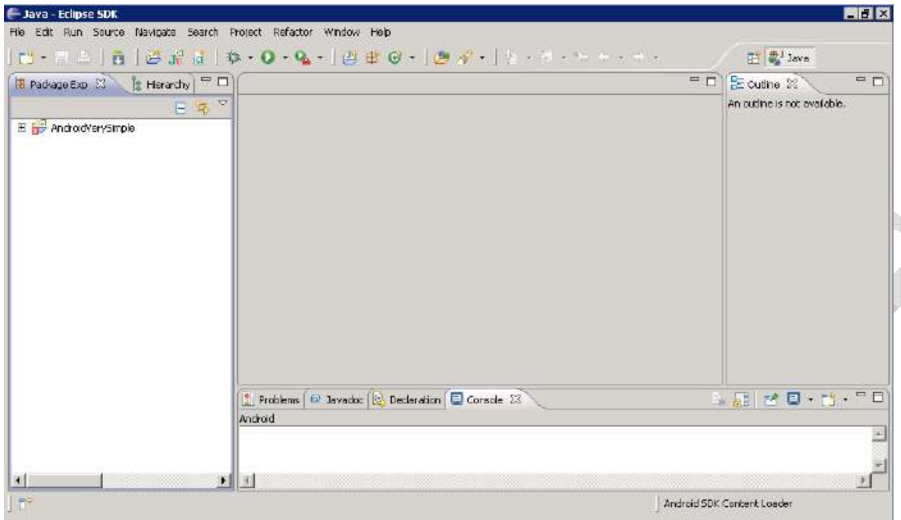


حيث سنقوم فيها بكتابة اسم المشروع و اسم التطبيق و اسم الحزمة و النشاط و تحديد المنصة الهدف و التي في حالتنا هذه هي الوحيدة (1.6) الآن قم بمليء الفراغات بالمعلومات التالية لتصبح لديك شاشة شبيهة بهذه :

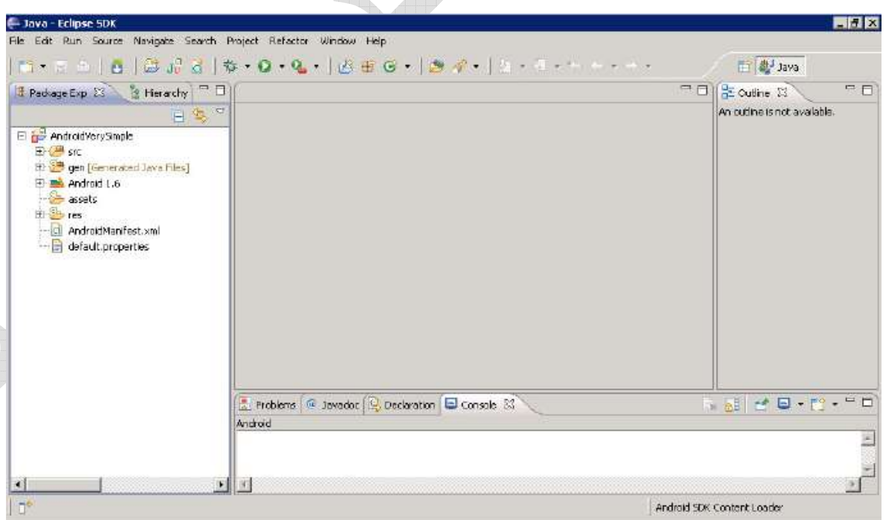


Project Nam: AndroidVerySimple
Build Target: 1.6
Application Name:VerySimple
Package name:com.book.VerySimple
Create Activity:Activity_VerySimple

ثم اضغط على زر إنهاء (Finish) ليتم فتح مشروع جديد بالشكل التالي

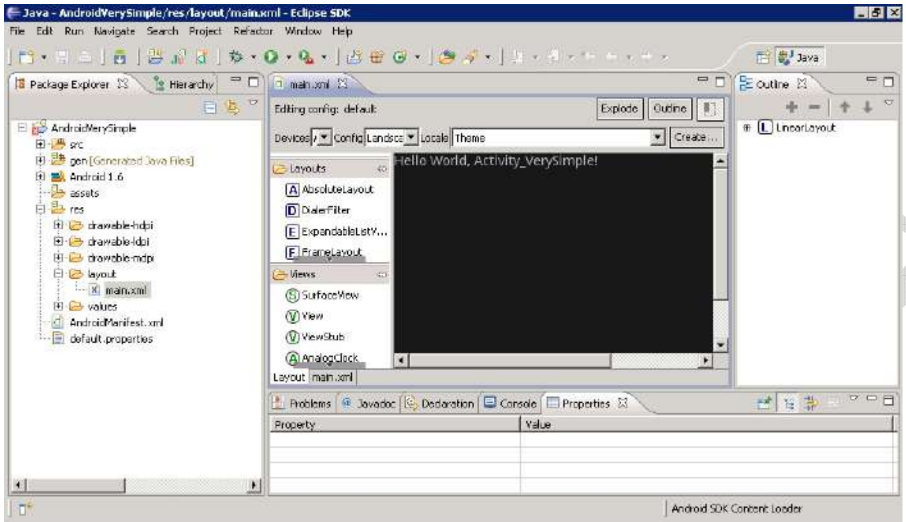


الآن دعنا نتعرف على ماذا قام المعالج ببنائه لنا و ذلك بسرد أبناء العنصر لتكون بالشكل التالي



الآن قم فتح العنصر (Res) ليظهر تحته (Layout) ومن ثم قم بالنقر مرتين على الملف (main.xml) لتظهر لك شاشة تطبيقك

الأولى و نرى أن أندرويد يجبرنا أن نرحب بالعالم بوضع (Hello World) و هذا ما سيظهر على الشاشة عند تشغيل البرنامج

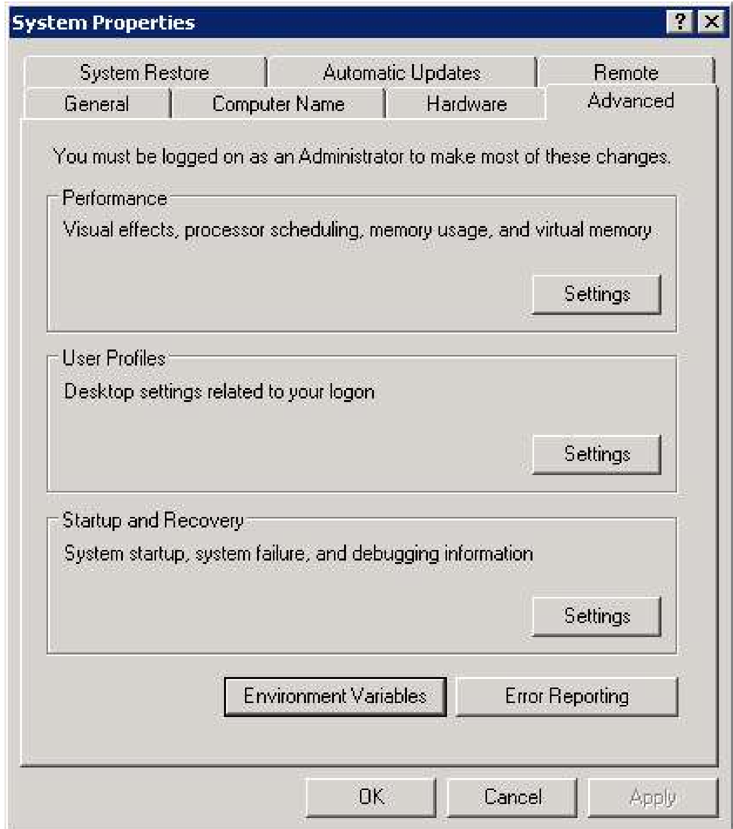


الآن لا داعي لتعرف أكثر من ذلك و لنقم بتشغيل تطبيقنا الأول و لكن في البداية علينا إعداد تطبيق المحاكي و لذا فلننتقل للخطوة التالية

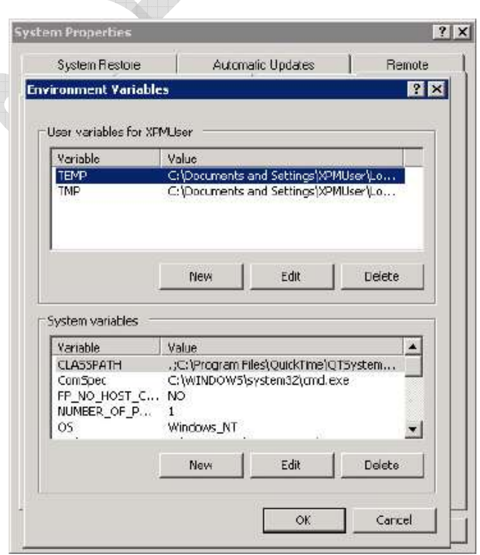
بناء مشروع جديد عن طريق سطر الأوامر

في حال لم تكن تملك (Eclipse) أو واجهت مشكلة أثناء عملية بناء معالج مشروع جديد فيمكنك دائما بناء المشروع عن طريق سطر الأوامر

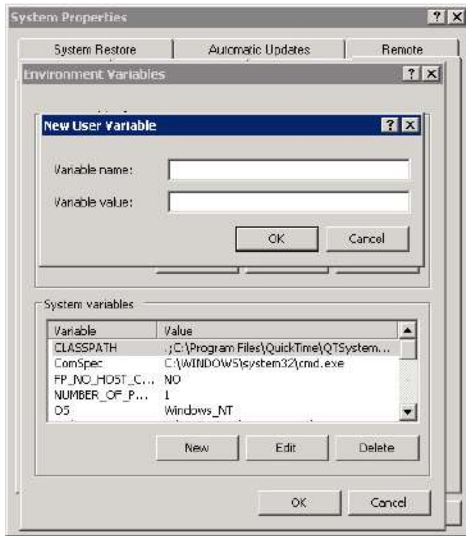
في البداية علينا تعريف مسار سطر الأوامر في النظام ليتعرف عليها عند طلبها في أي مسار و ذلك عن طريق الذهاب إلى جهاز الكمبيوتر من على سطح المكتب أو من القائمة إبداء و بالضغط على أيقونة جهاز الكمبيوتر بزر الفأرة الأيمن و من ثم الذهاب إلى خصائص (Properties) ومن ثم الذهاب إلى لسان تبويب خيارات متقدمة (Advanced) لتظهر لنا بالشكل التالي



ثم اضغط على زر (Environment Variables) وعندها ستظهر لنا شاشة المسارات كما الشكل التالي



في القسم العلوي و تحت بند (User Environment) اختر زر جديد (New) لتعريف مسار جديد لتظهر شاشة جديدة بالشكل التالي

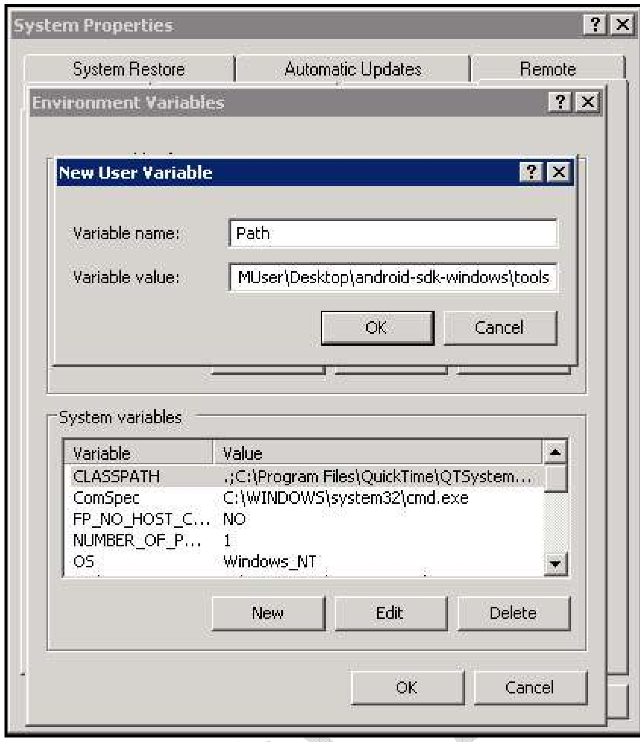


سنقوم بملء الشاشة بالمعلومات التالية

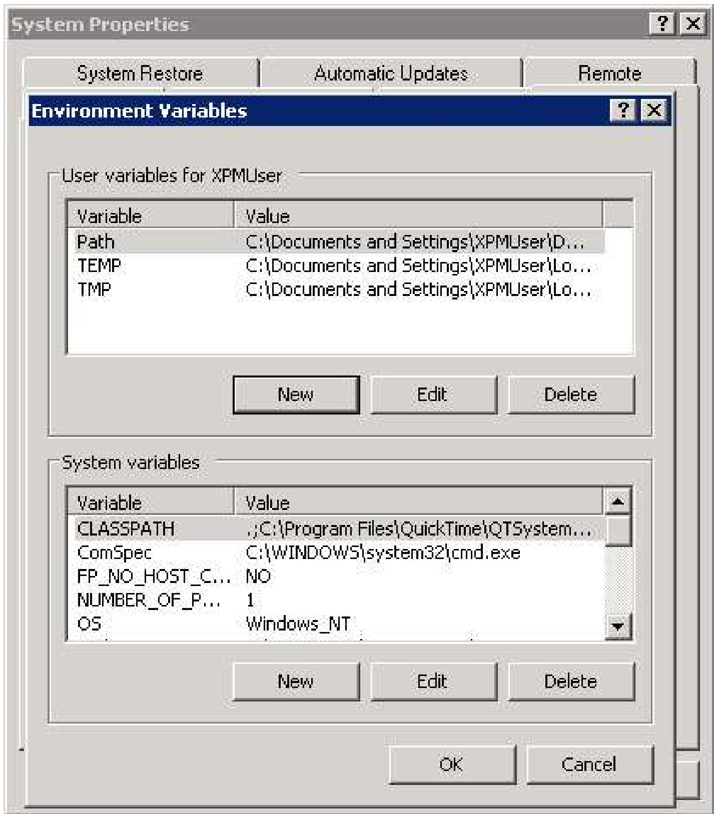
في مربع الاسم (Name) نكتب كلمة (Path) و في مربع القيمة (Variable Value) نضع مسار مجلد الأدوات الموجود داخل مجلد حزمة التطوير و الذي قمنا بفك ضغطه سابقا على سطح المكتب و لتكن بالشكل التالي

C:\Documents and Settings\UserName\Desktop\android-sdk-windows\tools

حيث أن (UserName) هو اسم المستخدم الحالي للنظام أو في حال قمت بفك ضغط الملف في أي مكان أخر فاذهب إليه و انسخ عنوان المسار كاملا مع مجلد (tools) و ضعه في القيمة

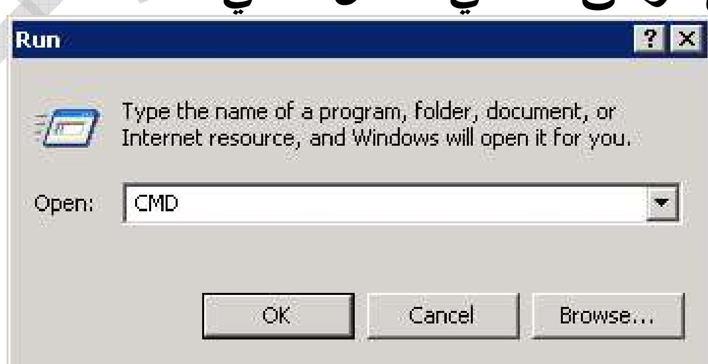


ثم اضغط موافق (OK) ليتم إضافة المسار الجديد ليكون بالشكل التالي

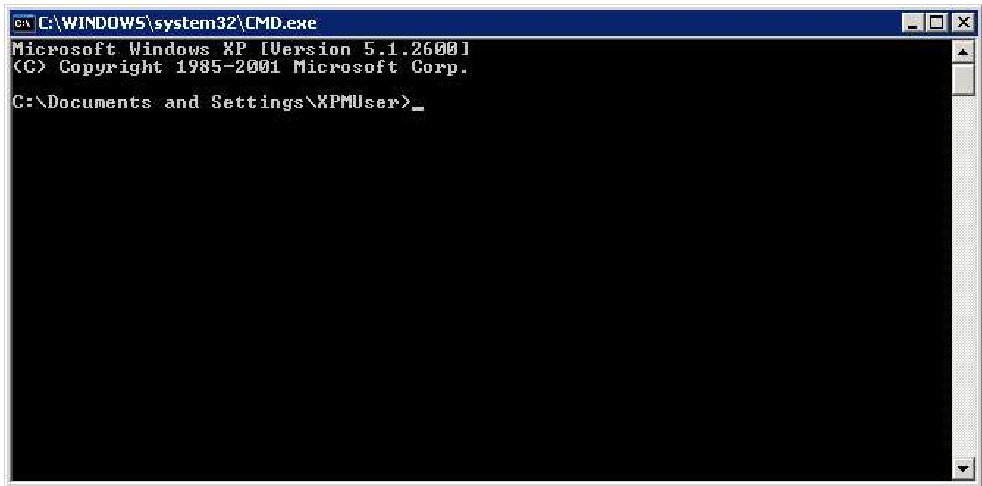


ثم بعد ذلك اضغط على زر موافق (OK) ثم مرة أخرى للشاشة الرئيسية اضغط موافق (OK)

الآن لتجرب استخدام الأوامر من سطر الأوامر قم بالذهاب إلى تشغيل من قائمة ابدأ (Start >> Run) و اكتب الأمر (CMD) ثم اضغط على موافق كما في الشكل التالي



عندما تضغط على زر موافق (OK) ستظهر لك شاشة سوداء و هي شاشة سطر الأوامر و الشبيهة بالشكل التالي

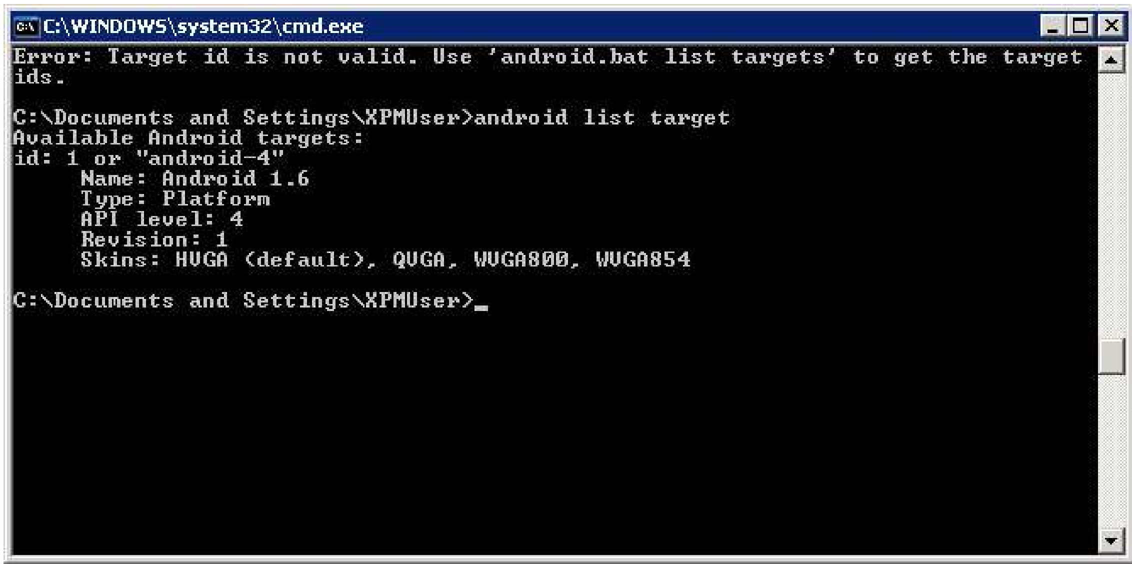


في البداية لنتعرف على المنصات المتواجدة في جهازنا و ترقيمها في الجهاز و ذلك عبر كتابة الأمر التالي

android list target

ثم اضغط زر الإدخال (Enter)

ستجد قائمة المنصات لديك كما في الشكل التالي



وفي حالتنا هذه لا يتوفر لدينا سوى منصة واحدة رقمها (1) و التي سنستخدمها لبناء المشروع الخاص بنا عبر سطر الأوامر الان لنجرب بناء مشروع جديد عن طريق سطر الأوامر فلذا علينا كتابة الأمر التالي

```
android create project -n AndroidVerySimple -p c:\1\  -k
com.book.VerySimple -a Activity_VerySimple -t 1
```

ملاحظة (سطر الأوامر السابق يكتب على سطر واحد و لكن النص هنا تم تقسيمه على سطرين لدواعي الكتاب)

حيث أن:

(AndroidVerySimple) اسم المشروع (Project Name) و نستخدم البارمتر التالي لتعريفها (n-)

(com.book.VerySimple) اسم الحزمة (Package Name)و نستخدم البارمتر التالي لتعريفها (-k)

(Activity_VerySimple) اسم النشاط (Activity Name) و نستخدم البارمتر التالي لتعريفها (-a)

(c:\1\) هو مسار المشروع الذي سوف يتم تخزينه على جهاز (Path)و نستخدم البارمتر التالي لتعريفها (-p)

(1) المنصة التي نستهدفها و التي استعرضنا رقمها في الخطوة السابقة (Target) و نستخدم البارمتر التالي لتعريفها (-t)

بعد الانتهاء من كتابة سطر الأمر كما هو بالكامل و الضغط على زر الإدخال (Enter) ستظهر لنا الشاشة التالية

```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\XPMUser>android create project -n AndroidVerySimple -p
 c:\1\  -k com.book.VerySimple -a Activity_VerySimple -t 1
Created project directory: c:\1
Created directory C:\1\src\com\book\VerySimple
Added file c:\1\src\com\book\VerySimple\Activity_VerySimple.java
Created directory C:\1\res
Created directory C:\1\bin
Created directory C:\1\libs
Created directory C:\1\res\values
Added file c:\1\res\values\strings.xml
Created directory C:\1\res\layout
Added file c:\1\res\layout\main.xml
Added file c:\1\AndroidManifest.xml
Added file c:\1\build.xml

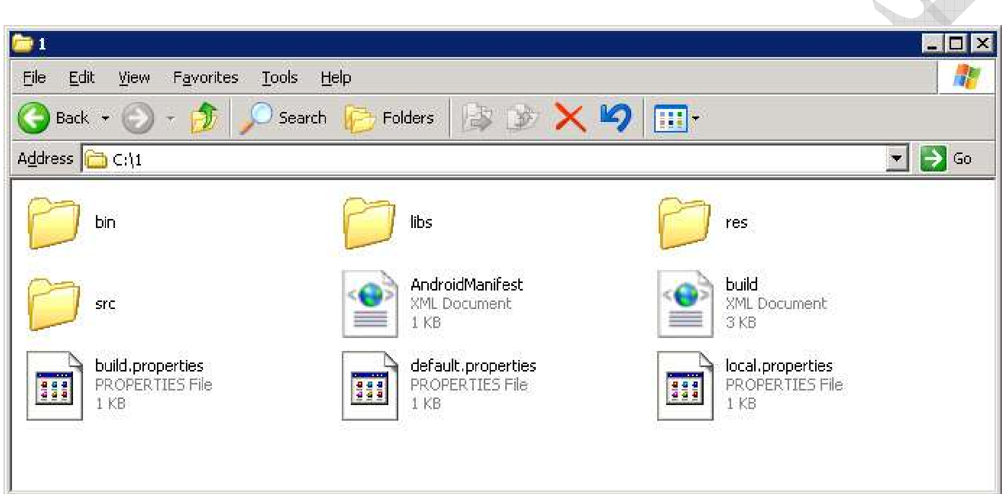
C:\Documents and Settings\XPMUser>_
```

يمكنك الآن تصفح ملفات المشروع داخل المجلد الذي قمنا بتحديده سابقا (c:\1) و الذي يمكنك أيضا اختيار نفس مساحة المشاريع

التي يستخدمها (Eclipse) في بناء المشاريع و التي تكون بالعادة على المسار التالي

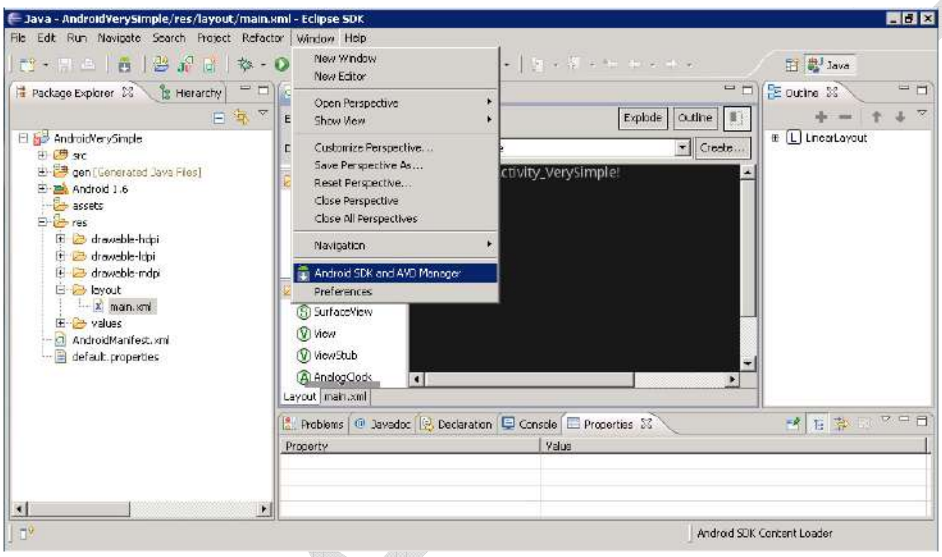
C:\Documents and Settings\UserName\workspace

لتستعرض الملفات التي قام المعالج عبر سطر الأوامر بإنشائها فلتذهب للمسار لتجد ملفات المشروع بالشكل التالي

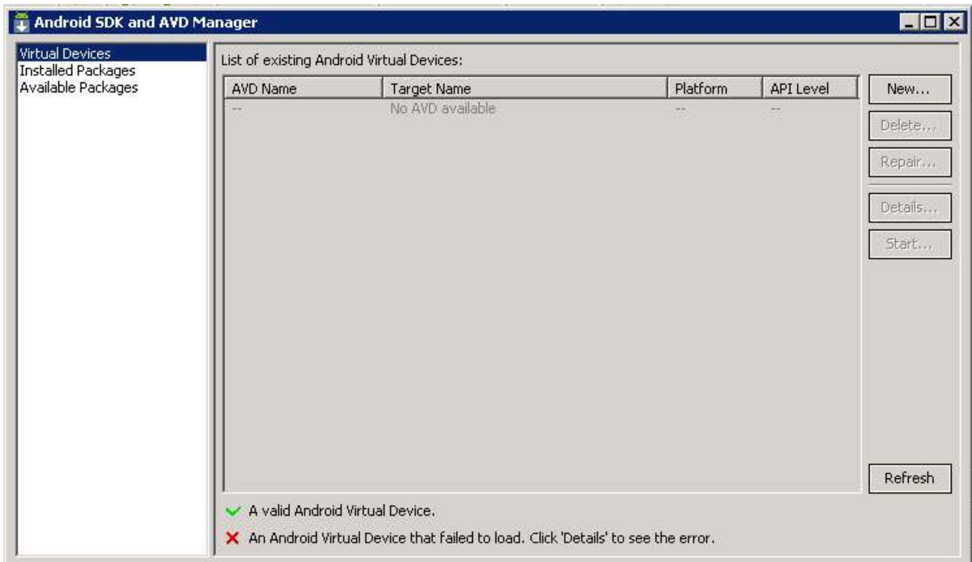


إعداد محاكى (Android)

من الشاشة الرئيسية لبيئة التطوير اذهب إلى القائمة (Window) و من ثم اختر العنصر (Android SDK and AVD Manager) كما هو موضح بالشاشة التالية

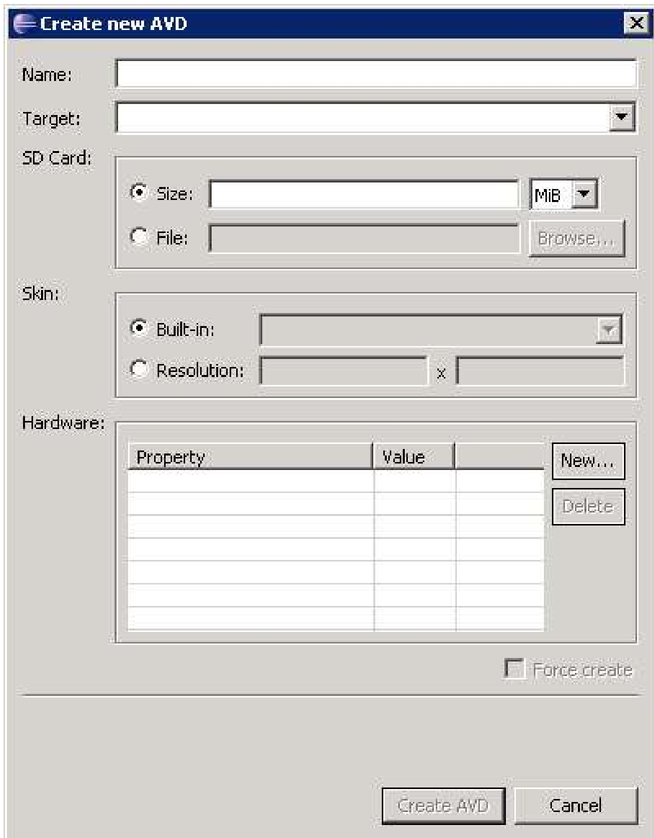


لتظهر لك الشاشة التالية



في حال واجهتك مشكلة ولم تفتح هذه الشاشة فلا تقلق فيمكنك دائما تشغيلها بشكل يدوي عبر الذهاب إلى المجلد الذي قمنا به بفرد حزمة التطوير و الذي يتواجد على سطح المكتب باسم (android-sdk-windows) و قم بتشغيل البرنامج (SDK Setup) و في حال لم يعمل معك أيضا فلا تقلق فهناك طريقة أخرى يمكن فيها أن نبني المحاكي عن طريق سطر الأوامر و التي سنتحدث عنها في أخر هذا الجزء.

لنفترض بأن كل ما لديك يعمل بشكل سليم و ظهرت الشاشة السابقة فلتذهب إلى الخيار (Virtual Devices) و من ثم اضغط على زر (New) في أعلم يمين الشاشة لتظهر لك شاشة إعداد جهاز افتراضي جديد.



في الشاشة السابقة سنقوم بتعريف بيانات الجهاز الافتراضي و ذلك بتحديد اسم خاص بنا و من ثم اختيار إصدارة نظام التشغيل مما قمنا بتحميلها سابقا أثناء عملية إعداد حزمة التطوير وفي حالتنا هذه لن يتواجد لدينا سوى الإصدارة (1.6) و من ثم نعرف الملحقات التي يدعمها الجهاز و منها كرت ذاكرة داخلي (SD) و بعض الملحقات الإضافية و التي سنتعرف على كيفية التعامل معها لاحقا.

الآن قم بكتابة اسم المحاكي (VM_Android1.6) و اختر إصدارة نظام التشغيل (1.6) و من ثم قم بتحديد حجم كرت الذاكرة بـ (500) ميجا و التي سنحتاجها لاحقا لكي نقوم بتحميل التطبيقات و الملفات عليها من سوق التطبيقات و التعرف على التطبيقات الموجودة و نوعيتها لنستطيع بناء تطبيقات مشابهة و بعد ذلك

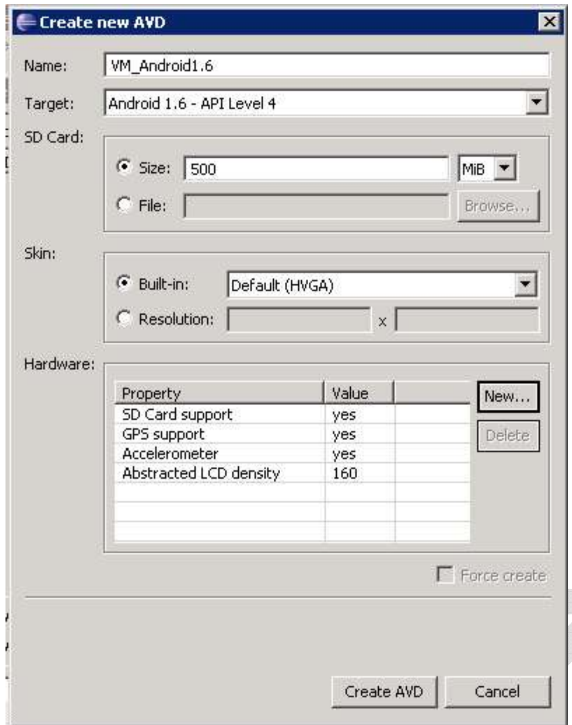
سنقوم بإضافة العتاد عن طريق القسم (Hardware) و اختيار زر (New) فتظهر لنا الشاشة التالية



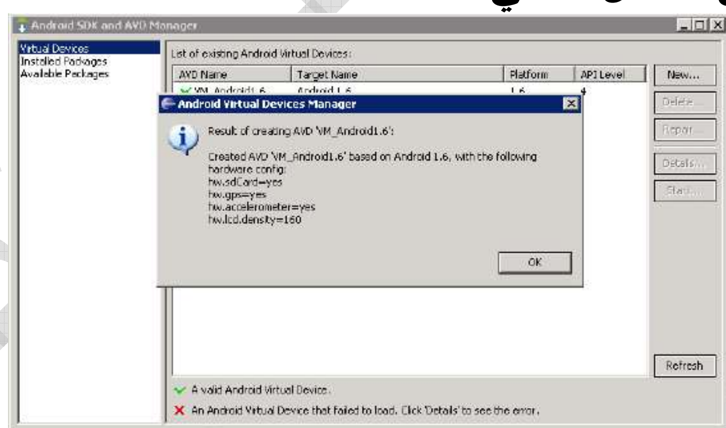
بما أننا اخترنا حجم كرت الذاكرة فعلينا دعم خاصية كرت الذاكرة و لذا سنضيف (SD Card Support) و من ثم سنضغط على زر (OK) و نكرر نفس العملية بالضغط على زر (New) لإضافة عتاد جديد و لنختر كل من

GPS Support
Accelerometer

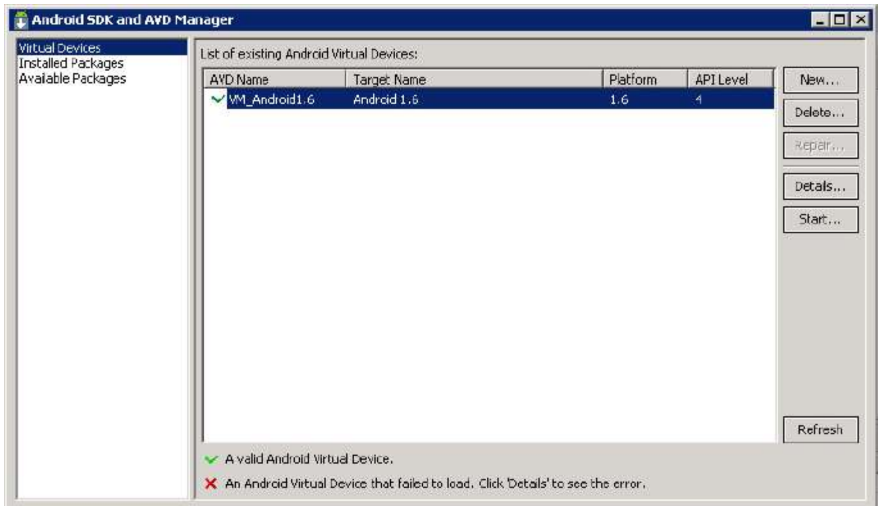
لتظهر لنا الشاشة النهائية بالشكل التالي



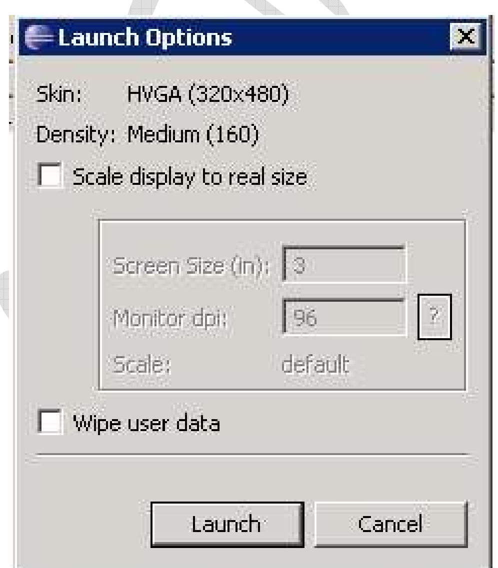
بعد ذلك قم بالضغط على زر (Create AVD) و انتظر قليلا ليتم بناء المحاكي و بعدها ستظهر لنا رسالة مفادها بأنه تم بناء لمحاكي بنجاح كما في الشكل التالي



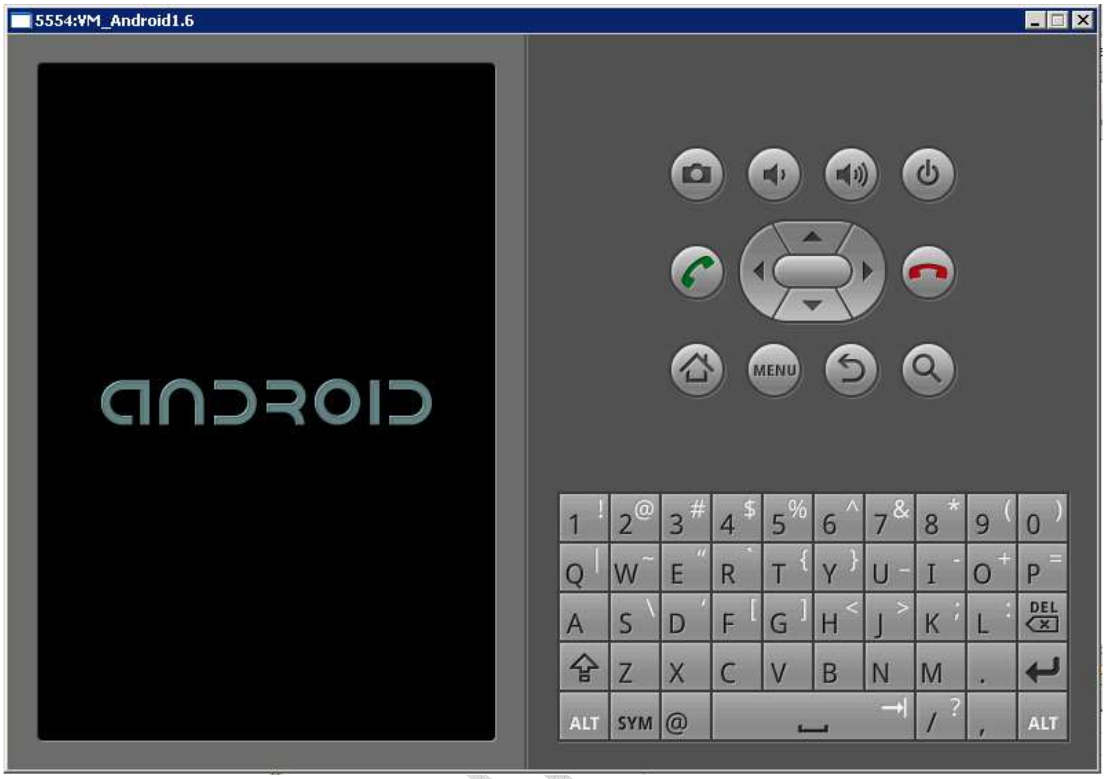
و بعد الضغط على زر موافق (OK) سيظهر الجهاز ضمن قائمة الأجهزة كما في الشكل التالي



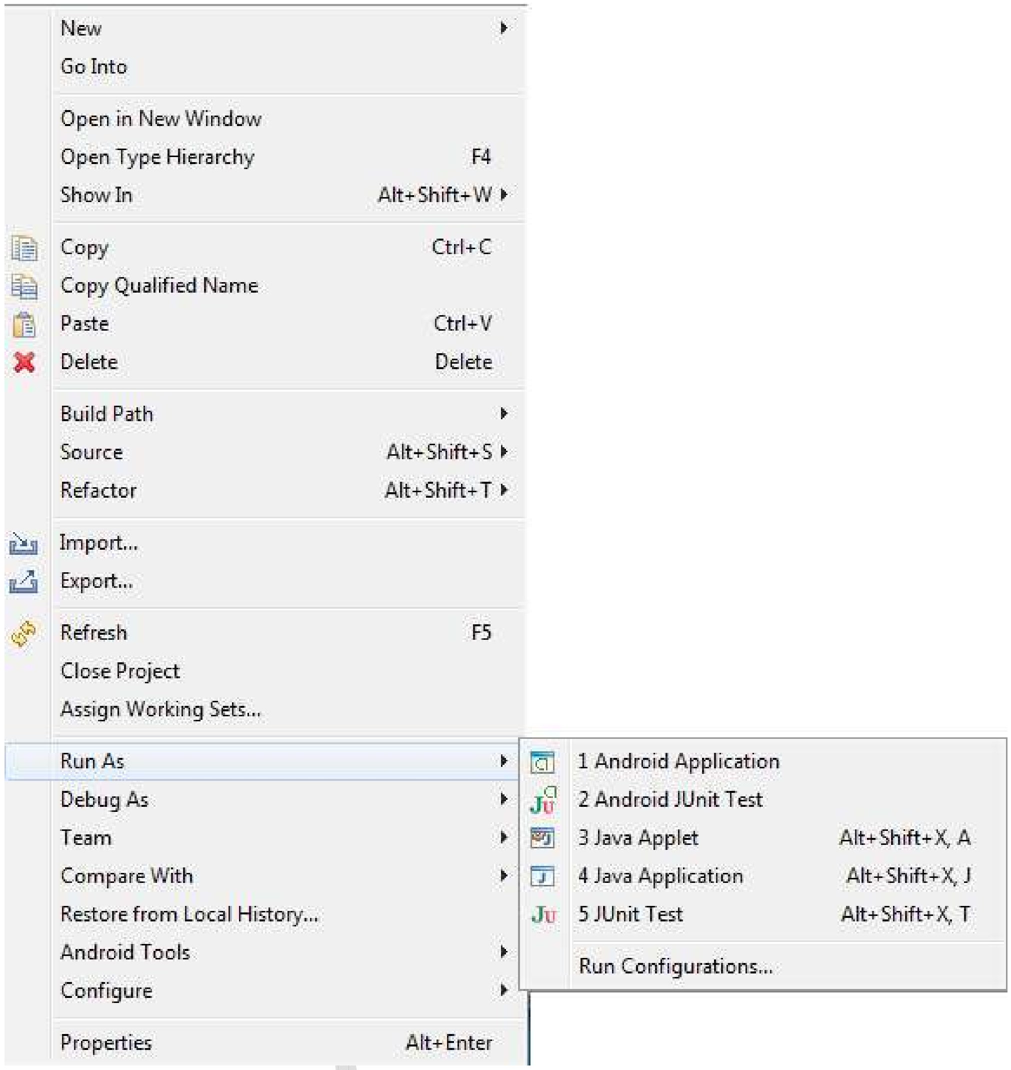
الان لتشغيل الجهاز ما عليك سوى تحديده و من ثم الضغط على زر (Start) و عندها ستظهر لنا شاشة بالشكل التالي تجاهلها للوقت الحالي و اضغط (Launch)



الان سيبدأ المحاكي بالعمل ليظهر لنا بالشكل التالي



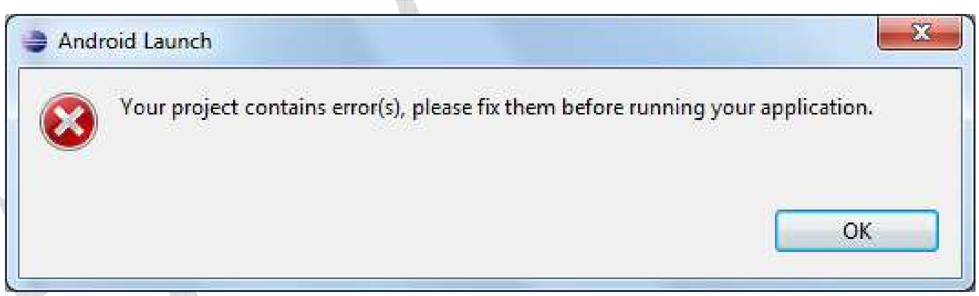
بعد أن شغلنا المحاكي و الذي يحاج بعض الوقت لإقلاع نظام التشغيل الداخلي إذا أردت أن تتجول في الجهاز فلك ذلك و لكن بعد ذلك فلنعود إلى بيئة التطوير لنقوم بتشغيل برنامجنا الأول و ذلك عبر الذهاب إلى اسم المشروع في رأس الشجرة و الضغط بزر الفأرة الأيمن و من ثم اختيار (Run As) ومن ثم من القائمة الفرعية اختر (Android Application)



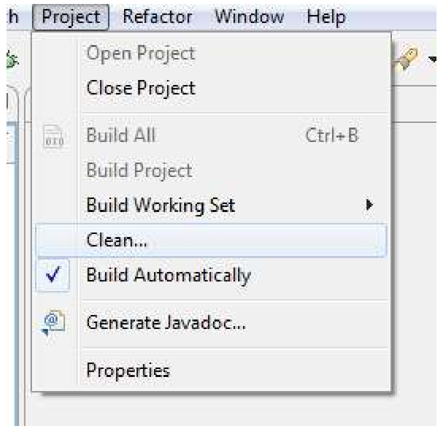
و ستجد أن التطبيق الخاص بك سيظهر في المحاكي و هي عبارة عن شاشة فقط تحتوي على سطر وحيد (HelloWorld)



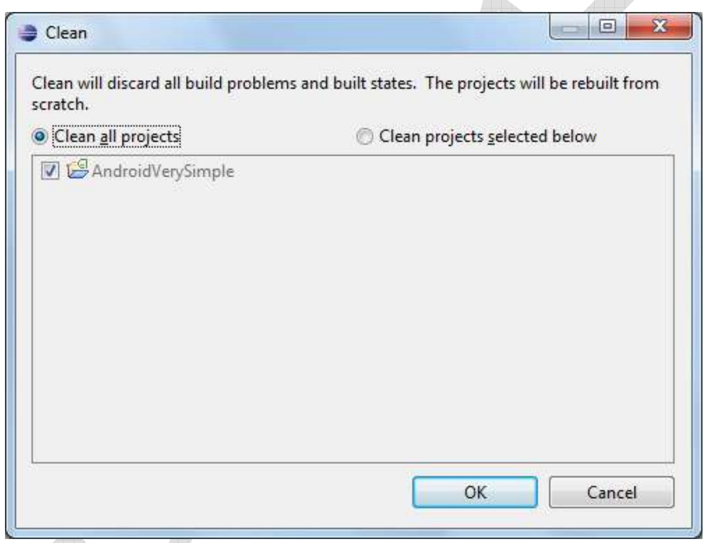
في حال واجهت مشكلة عند محاولة تشغيل تطبيقك و ذلك بسبب أنه أحيانا معالج إعداد المشروع لا يقوم ببناء كامل المراجع في ملفات المشروع المتنوعة وستظهر لك رسالة خطأ شبيهة بالتالية



فلذلك إما عليك أن تغلق المشروع و تعيد فتحه أو تقوم بالذهاب إلى القائمة (Project) ومن ثم اختيار خيار (Clean) كما في الصورة التالية



و بعد ذلك ستظهر لك شاشة كما في الشكل التالي اضغط على زر موافق (OK)

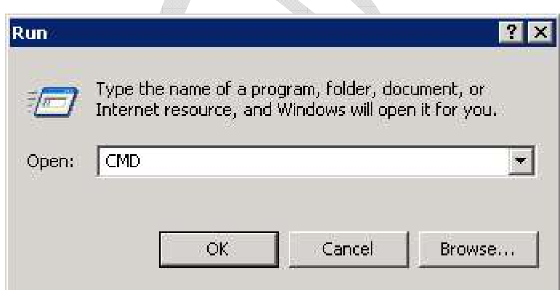


ستلاحظ أنه كانت هناك علامة (X) صغيرة حمراء كانت على اسم المشروع في أعلى الشجرة و بعد هذه العملية ستختفي هذه العلامة و عليك الآن أن تكرر الخطوة السابقة لتشغيل تطبيقك على المحاكي.

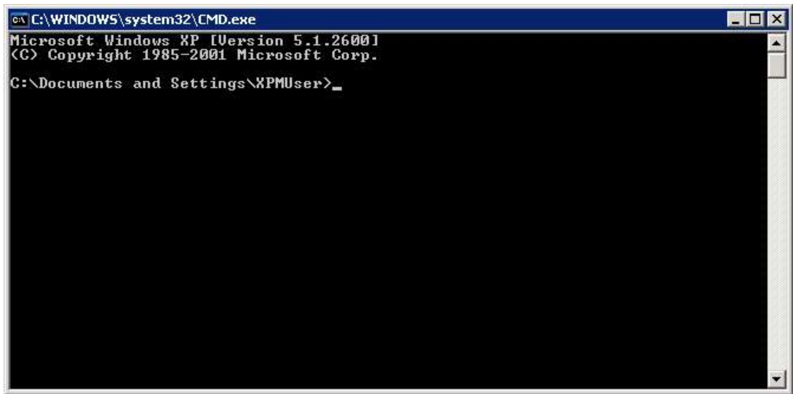
التعامل مع الجهاز الافتراضى عن طريق سطر الأوامر

كما ذكرنا سابقا كيف يمكن إنشاء مشروع جديد عن طريق سطر الأوامر دعنا نتعلم كيفية التعامل مع المحاكي عن طريق سطر الأوامر و لكن من المفروض أنك قمت بإضافة (Environment Variables) كما تعلمنا ذلك عندما شرحنا كيف تبدأ مشروع جديد و في حال قمت بذلك سابقا فلا حاجة لتكرار الأمر و عليك الذهاب مباشرة لتشغيل سطر الأوامر

قم بالذهاب إلى تشغيل من قائمة ابدأ (Start >> Run) و اكتب الأمر (CMD) ثم اضغط على موافق كما في الشكل التالي



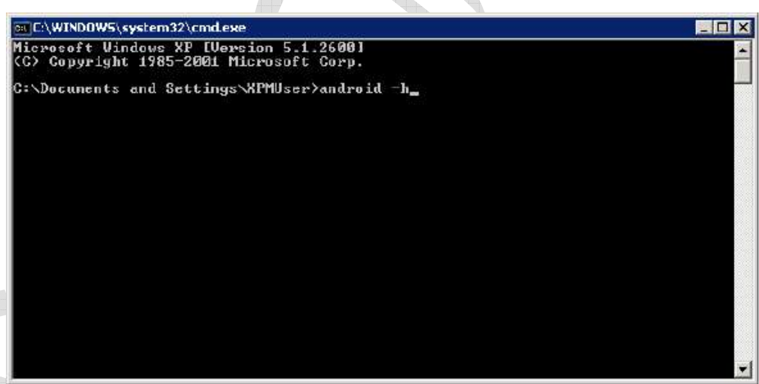
عندما تضغط على زر موافق (OK) ستظهر لك شاشة سوداء و هي شاشة سطر الأوامر و الشبيهة بالشكل التالي



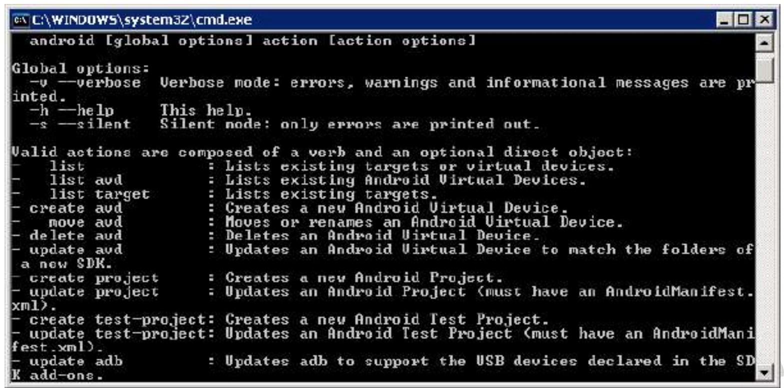
الآن قم بتجربة كتابة الأمر التالي لاكتشاف ملف المساعدة و البارامتر المختلفة التي يمكن استخدامها لإنشاء جهاز

Android –h

و من ثم اضغط على زر الإدخال (enter)



(android -h) حيث أن المتغير (h) الذي قمنا بتمريره للأمر (android) يطلب منه عرض ملف المساعدة و الذي يشرح كل المتغيرات الممكنة و ستظهر لنا قائمة كبيرة يمكن الإطلاع عليها و هي شبيهة بهذه الشاشة

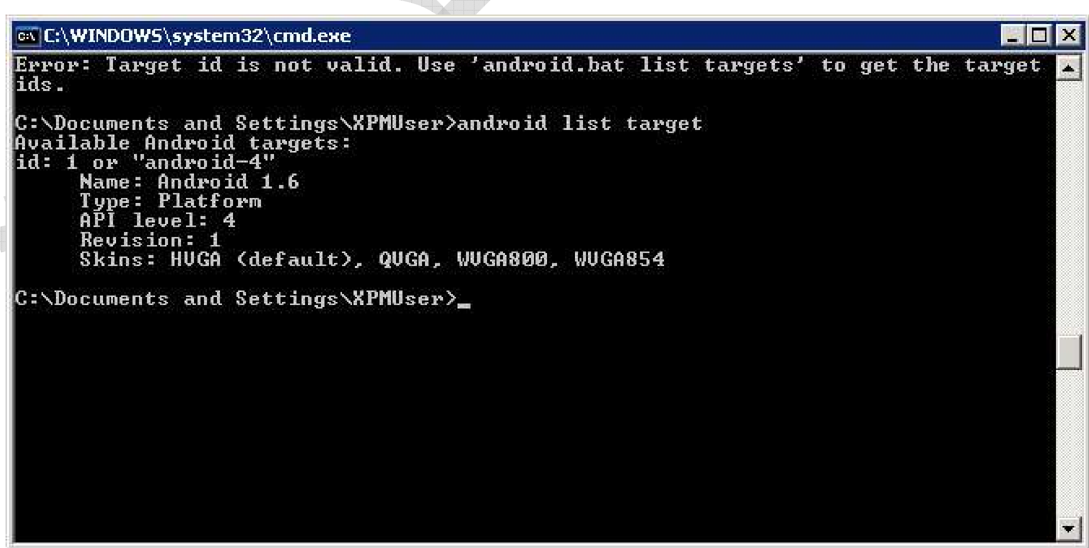


مثال في حال أردنا إضافة جهاز افتراضي جديد جرب السطر التالي أولا لمعرفة المنصات و أرقامها لديك

android list target

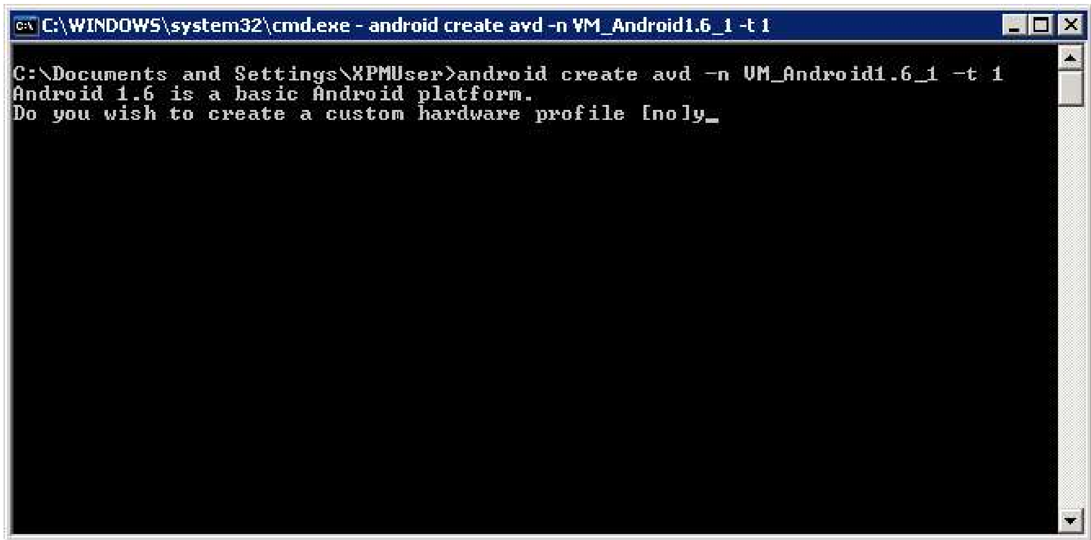
ثم اضغط زر الإدخال (Enter)

ستجد قائمة المنصات لديك كما في الشكل التالي

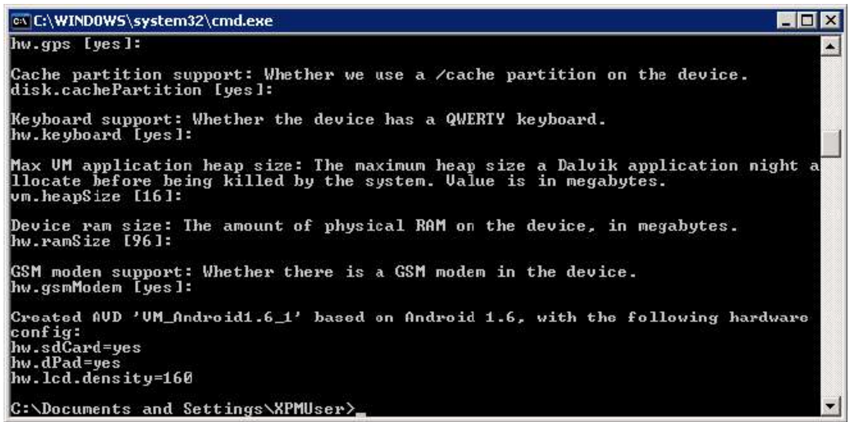


وفي حالتنا هذه لا يتوفر لدينا سوى منصة واحدة رقمها (1) و الذي سنستخدمه لبناء المحاكي الخاص بنا عبر سطر الأوامر لكتابة السطر التالي ثم الضغط على زر الادخال (Enter)

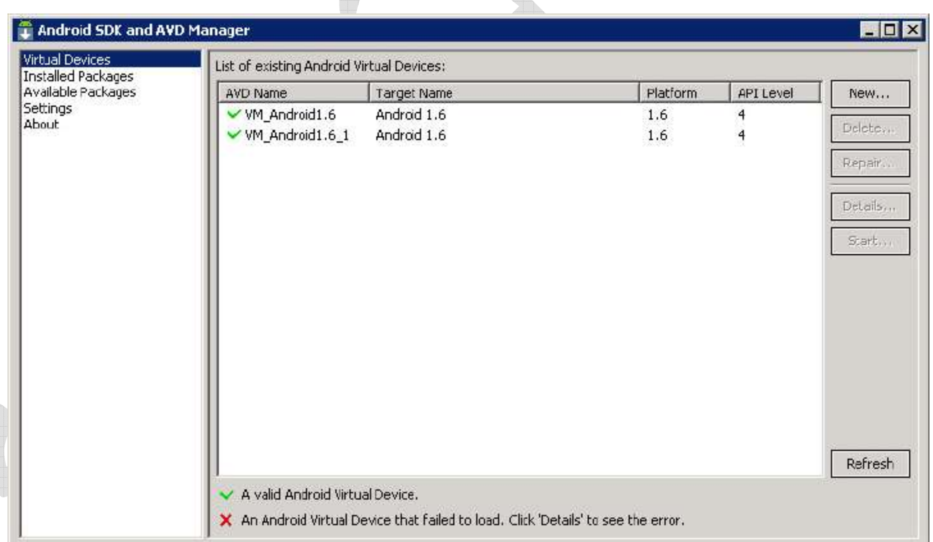
android create avd -n VM_Android1.6 -t 1



حيث أن (VM_Android1.6) هو اسم المحاكي و (create avd -n) يعني إنشاء جهاز افتراضي جديد و البارمتر (-t) يحدد الواجهة المستخدمة و في حالتنا هي (1) ليظهر لنا المعالج بعد ذلك ليسألنا بعض الأسئلة عن العتاد الذي نود إضافته و الذي سنجيب عليه بـ (yes) أو (No) فقط أو بالضغط على زر الإدخال (Enter) لأخذ القيمة الافتراضية التي تكون ضمن القوسين مثال [yes]

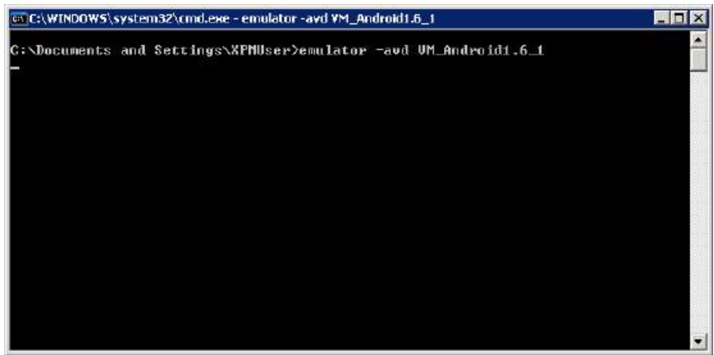


ملاحظة: إن كنت قد قمت ببناء الجهاز المحاكي (VM_Android1.6) عبر المعالج الرسومي سابقا و جربت القيام بنفس الشيء عبر سطر الأوامر فستأتيك رسالة بأن الاسم قد تم حجزه لاحقا و تحتاج إلى تغيير اسمه مثلا ليكون (VM_Android1.6_1)



بعد الانتهاء من إنشاء المحاكي و لتشغيله قم بكتابة الأمر التالي

emulator -avd VM_Android1.6_1



ليظهر لديك المحاكي

ملاحظة: في حال قمت بإغلاق نافذة سطر الأوامر فأنك ستقوم بإغلاق المحاكي تباعا لان الأمر مازال تحت التشغيل حتى ينتهي و لذا في حال قمت بتشغيل المحاكي عن طريق سطر الأوامر فانك ستحتاج للإبقاء على شاشة سطر الأوامر تعمل في الخلفية أو قم بالذهاب إلى حزمة التطوير و تشغيل (SDK Setup) لتجد بأن المحاكي الجديد الذي قمنا بإنشائه ضمن سطر الأوامر موجود بالإضافة للسابق ضمن قائمة الأجهزة الافتراضية.

إضافة اللغة العربية إلى المحاكي (يتوجب عليك الحصول على ملفات الخطوط: DroidSans-Bold.ttf, DroidSansFallback.ttf, DroidSans.ttf)

adb remount
adb shell rm /system/fonts/*

adb push c:\fontss\DroidSans-Bold.ttf /system/fonts/
adb push c:\fontss\DroidSansFallback.ttf /system/fonts/
adb push c:\fontss\DroidSans.ttf /system/fonts/

في حال ظهور رسالة خطأ تفيد بعدم وجود مساحة كافية و يمكنك حذف بعض التطبيقات، مثلا:
adb shell rm /system/app/AlarmClock.apk

تنصيب برنامج من سطر الأوامر

adb install c:\Vending.apk

المحاكي و العالم الخارجي

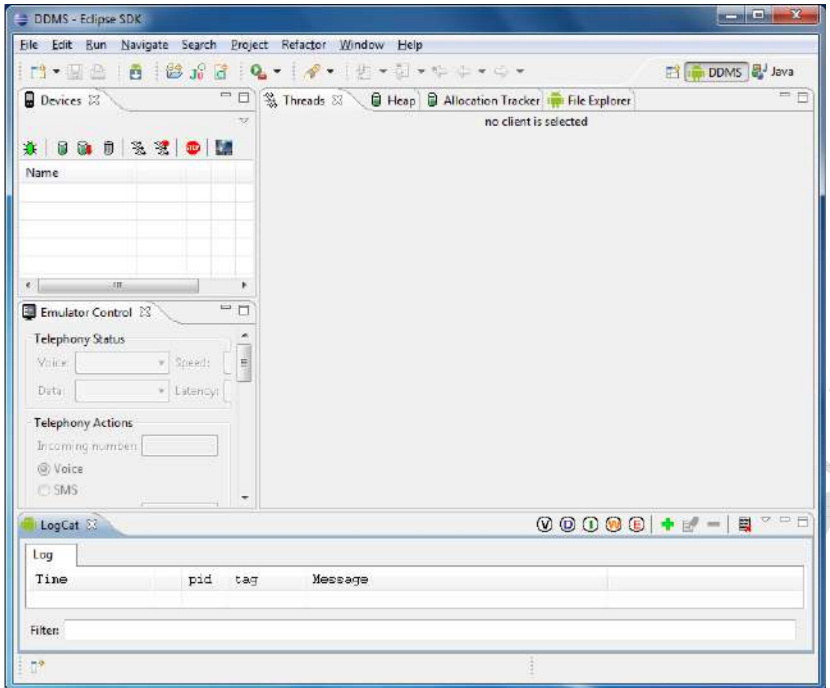
لنتعرف الآن على أداة جديدة موجودة ضمن بيئة التطوير تتيح لنا التخاطب مع المحاكي و الوصول إلى ملفاته و خصائصه و التعامل معه و كأنه جهاز حقيقي لكي نستطيع تجربة كل خصائص الجهاز و هذا بالتأكيد لا يغني عن تجربة تطبيقك النهائي على جهاز حقيقي لتعرف كيفية عمله

أداة (DDMS)

أفترض حاليا أنك قد قمت بتشغيل بيئة التطوير (Eclipse) و علينا تشغيل أداة (DDMS) و ذلك من أعلى يمين الشاشة قم بالضغط على زر الإضافة (+) لتظهر لنا قائمة كما الشكل التالي

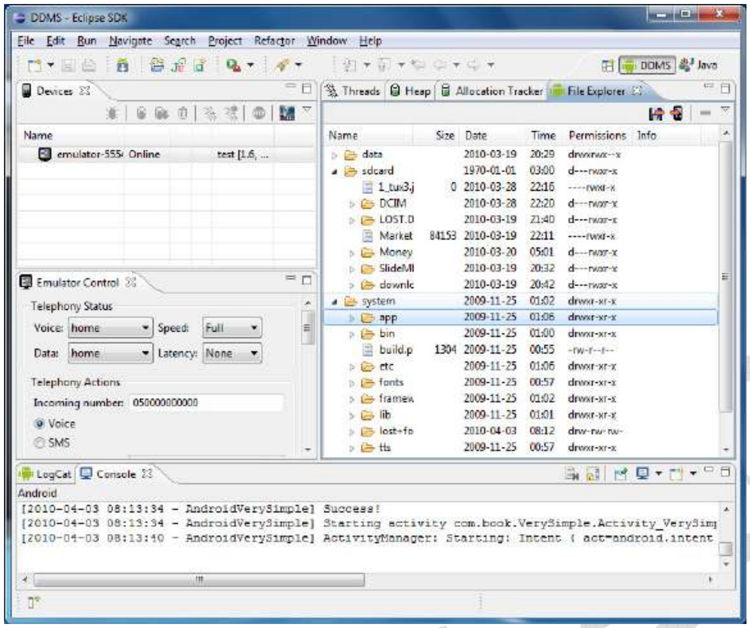


قم باختيار (DDMS) لتصبح ضمن محتويات الشاشة و يمكن الوصول لها مباشرة و لتظهر لنا الشاشة التالية



الان ستلاحظ بأن (DDMS) أصبحت في الأعلى و التي يمكن التحويل ما بينها و ما بين بيئة التطوير (Java) لسهولة الوصول و التنقل

الخطوة التالية تقتضي بأن نقوم بتشغيل المحاكي بأي من الطرق السابقة التي قمنا بشرحها ليتعرف عليها (DDMS) و يمكننا من التواصل معها لتظهر لنا شاشة شبيهة بالشكل التالي بعد تشغيل المحاكي



كما ترى فإنه يمكنك الوصول لملفات الجهاز بالكامل و كذلك الحال لكرت الذاكرة الداخلي من الجهة اليمنى العلوية (File Explorer) و معرفة العمليات التي تعمل في الخلفية (Threads) أو (Heap) أو (Allocation Tracker)

يمكنك من قائمة (File Explorer) اختيار المجلد مثلا و ليكن كرت الذاكرة (sd card) و من ثم من أعلى القائمة اختيار زر هاتف الجوال لسحب ملف من جهاز الكمبيوتر إلى المحاكي أو اختيار زر الحفظ لنسخ ملف من المحاكي إلى جهاز الكمبيوتر.
تنويه: قم بالتعرف على هيكلية ملفات النظام و قم بنسخ الملفات على جهازك و اعرف كيفية بنيتها لتعرف لاحقا كيف يتعامل النظام معها أو حال في أردت تغيرها مثلا (تغيير ملفات الرنين) أو صورة الخلفية.

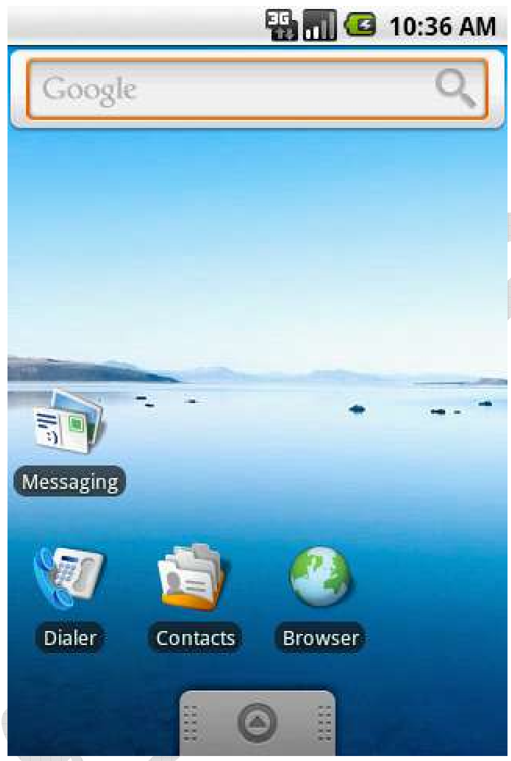
في الجهة اليسرى ستجد قائمة بالأجهزة (Devices) التي تعمل حاليا حتى لو قمت بشبك جهاز فعلي و قمت بتعريف (USB

Driver) و الموجود ضمن حزمة التطوير (SDK) فإنك ستجد الجهاز بهذه القائمة و يمكن الوصول لملفاته الداخلية و التعامل معه كما لو أنه محاكي.

تحت قائمة الأجهزة توجد قائمة التحكم بالمحاكي (Emulator Control) و التي تمكنك من التخاطب مع الجهاز كما لو أنه فعلي فعلى سبيل المثال تستطيع كتابة رقم هاتف و الاتصال فيظهر على شاشة المحاكي بأن هناك اتصال وارد من الرقم الذي كتبته و كذلك الحال في حال أردت إرسال رسالة نصية (SMS) أو أردت أن ترسل مكان عبر التوضع العالمي (GPS) عبر تمرير خطي الطول و العرض (Longitude) و (latitude) في حال كان تطبيق يعتمد مثلا على جهاز (GPS) .

تدريب

ما رأيك الآن لو تقوم بالعمل على المحاكي قليلا و تحميل بعض التطبيقات المجانية من سوق البرامج لتتعرف أكثر على هذه البرامج



من شاشة جهاز المحاكي قم بالذهاب إلى متصفح الانترنت (Browser) كما في الشكل التالي

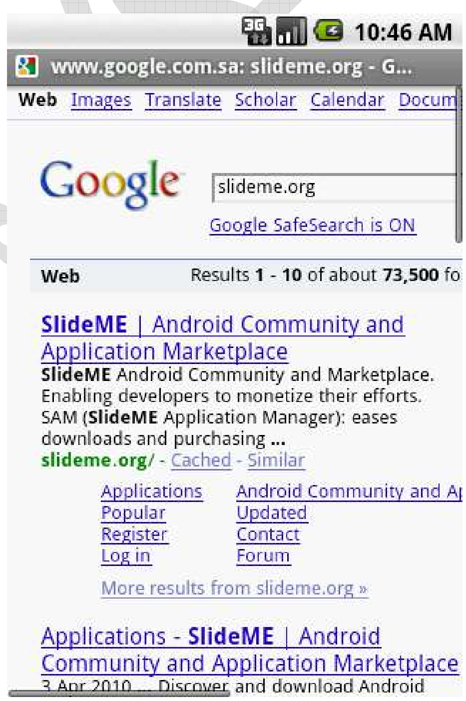


قد تظهر لك الشاشة بهذا الشكل و ذلك بسبب أن المحاكي يقوم بالتعرف على إعدادات الجهاز لديك و يرسلها للموقع و الذي يأتي بالواجهة العربية و بما أن المحاكي لا يحتوي على الخطوط العربية حتى الآن و لذا قم باختيار (Google.com in English) للتحول إلى الواجهة الانجليزي

تنويه: المحاكي يقوم بالحصول على الانترنت مباشرة من جهازك فلذا لا بد من أن يكون جهاز الكمبيوتر لديك متصل بالانترنت



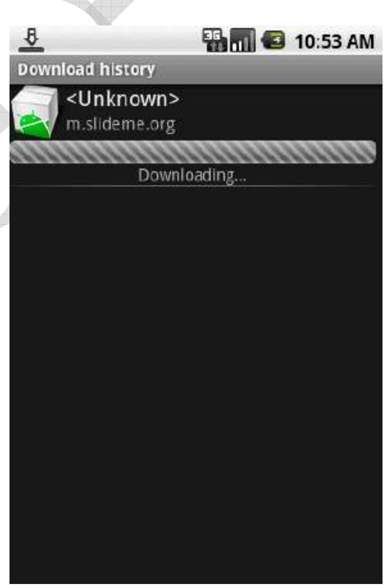
و قم بكتابة العنوان التالي للذهاب إلى أحد مواقع سوق البرامج العامة (slideme.org)



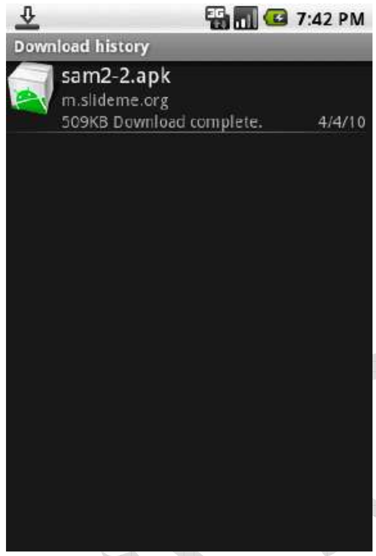
الآن اذهب إلى الموقع في القائمة



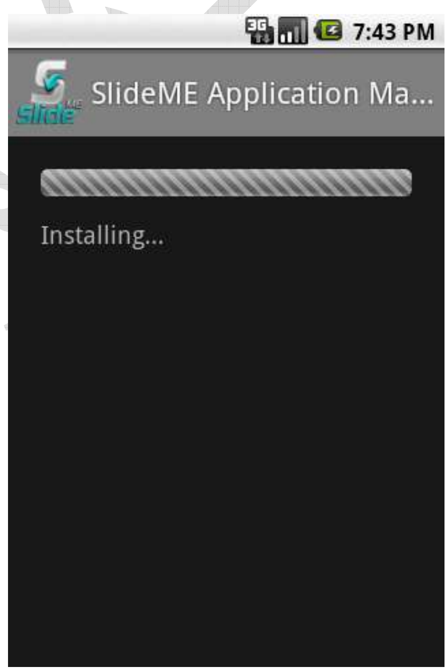
وقم بالضغط على الزر الأحمر (SAM v2.42) ليبدأ في تحميل البرنامج على جهازك



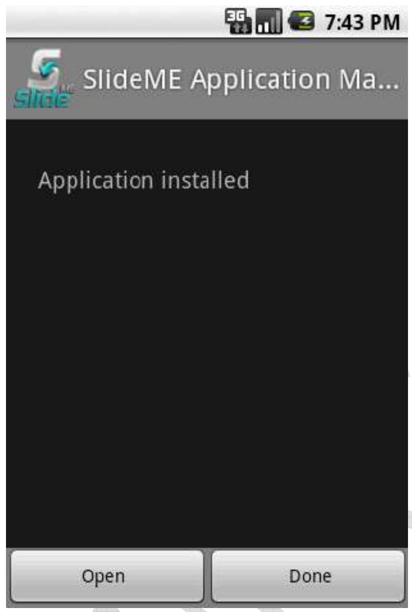
بعد انتهاء تحميل البرنامج بعمل تنصيب له و ذلك بالضغط على صورة البرنامج



بعد ذلك ستبدأ عملية تحميل التطبيق على جهازك



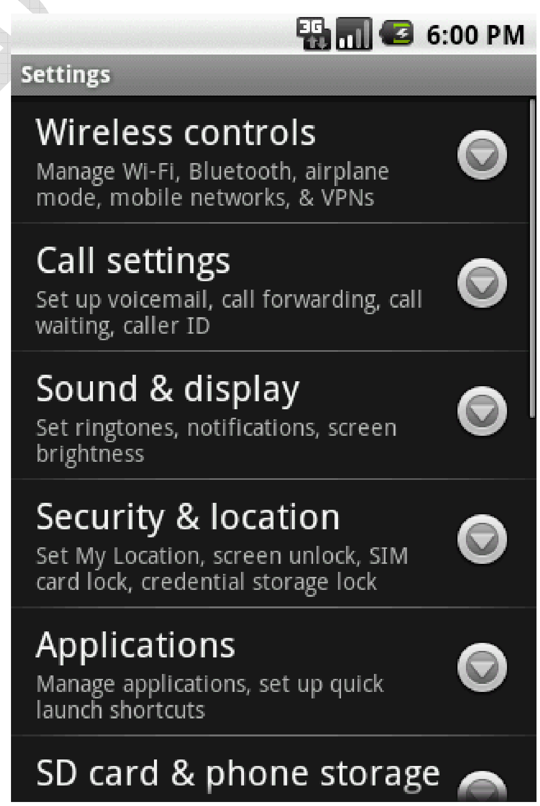
بعد الانتهاء من عملية التحميل عليك أن تقوم بالضغط على زر (Done) لانتهاء عملية التحميل



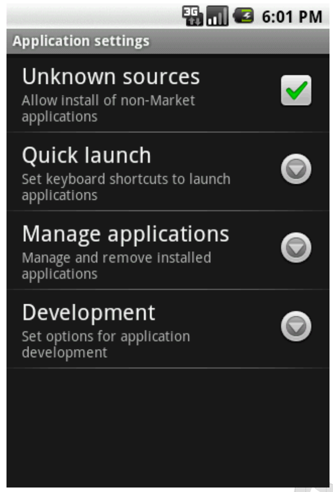
في حال لم تتمكن من عملية التحميل فعليك أن تقوم بعمل تعديل بسيط في إعدادات النظام للسماح بالتطبيقات الغير موقعه بالتحميل على جهازك و بجميع الأحوال أنت بحاجة لتفعيل هذا الخيار لتستطيع من تحميل أغلب التطبيقات الموجودة في سوق (Slide Me) لاحقا على جهازك و للقيام بذلك قم بالضغط على زر (Menu) و آنت على الشاشة الرئيسية لتظهر لك الشاشة التالية



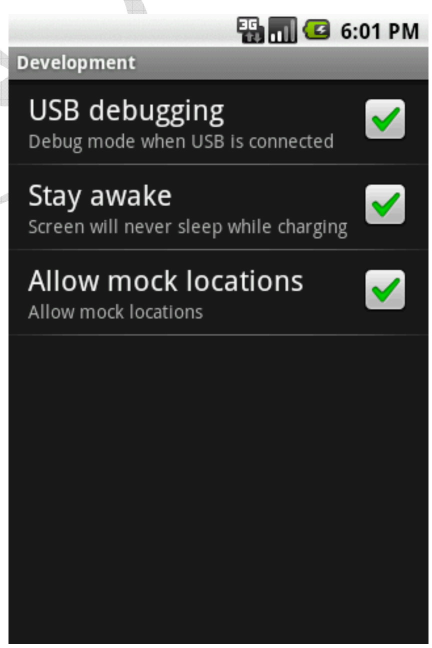
عند الضغط على زر (Settings) ستظهر لك شاشة بخيارات الإعدادات كالتالي



قم بالضغط على خيار (Applications) للوصول إلى القائمة الفرعية الخاصة بها كهذه



قم بتمكين الخيار الاول (Unknown sources) إن لم يكن كذلك ليصبح كما الشاشة التالية و كذلك الحال تحتاج لتطبيق ذلك في جهازك الفعلي إن أردت التجربة عليه و بعد ذلك اضغط أيضا على الخيار(Development) لتظهر لك شاشة شبيهة بالتالية



قم بعبرها بتمكين الثلاث خيارات في حال لم تكن كذلك و هي الخاصة بتمكين التعامل مع (DDMS) و تسهيل عملية متابعة الأخطاء و البرمجة على الأجهزة و كذلك الحال تحتاج إلى عمل نفس الخطوات على جهازك الفعلي في حال أردت التجربة عليه
الان عبر سهم الرجوع قم بالضغط عليه عدة مرات للعودة إلى الشاشة الرئيسية و حفظ الإعدادات التي قمنا بها

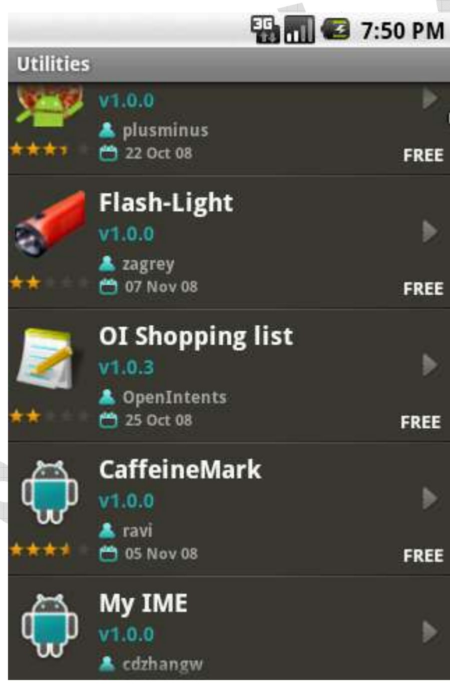
الان من قائمة البرامج قم بتشغيل برنامج (Slide Me) لتظهر الشاشة التالية و التي تحتوي على تصنيفات فرعية لأنواع التطبيقات الموجودة و عند تشغيل البرنامج لأول مرة ستظهر لك شاشة اتفاقية الاستخدام قم بالموافقة عليها لتظهر لك شاشة البرنامج



قم باستكشاف أنواع التطبيقات في كل قسم و حمل منها ما تشاء فمنها ما هو مجاني و منها ما هو مدفوع و في حال أردت تجربة التطبيقات المدفوعة فأنت تحتاج للتسجيل في موقع

(slideme.org) و تخزين بيانات بطاقاتك الائتمانية و تأكيد ملكيتك لها عبر عملية تتم أثناء التسجيل مما يتيح لك الشراء مباشرة من التطبيق و تقييد المبالغ مباشرة من الموقع على حساب بطاقتك الائتمانية دون الحاجة لمعرفة الطرف الثالث لبيانات بطاقة الائتمانية أو لإدخال بياناتك مرة أخرى في أي مكان

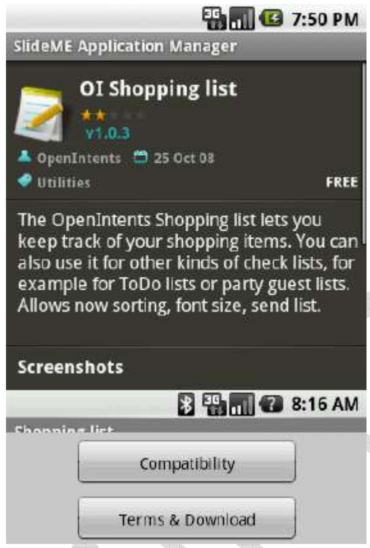
فلنجرب الآن تحميل تطبيق من القائمة (Utilities) و هو تطبيق قائمة المشتريات (OI Shopping List) للتعرف عليه و من ثم في الأجزاء القادمة لنقوم ببناء تطبيق مشابه بقائمة المهام



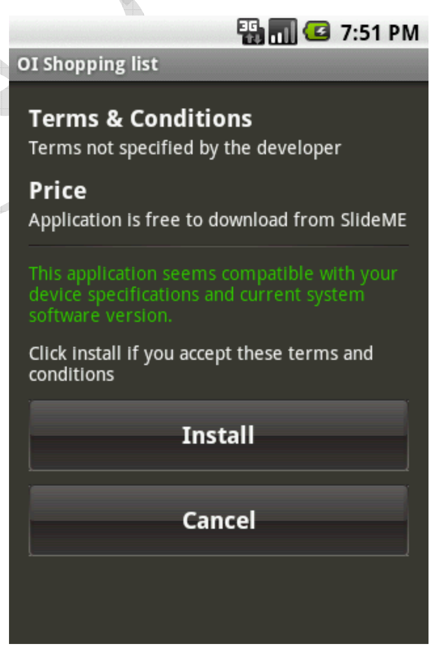
من القائمة (Utilities) ابحث عن البرنامج (OI Shopping List)

اضغط عليه لتظهر لك شاشة بمعلومات البرنامج و التي يمكنك أن تتعرف فيها أكثر عن معلومات التطبيق بالإضافة إلى صور عن

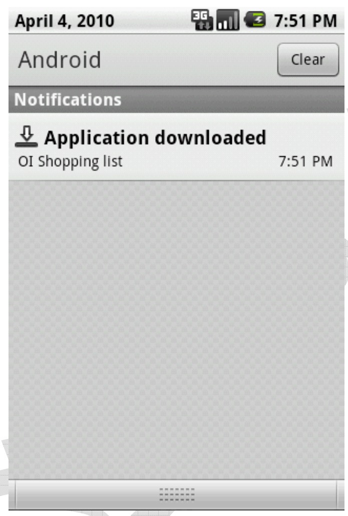
واجهة البرنامج و سعره و ما إلى ذلك من أمور مع العلم أن هذا البرنامج مجاني بل و يمكن تحميل الشفرة المصدرية الخاصة به من موقع المبرمج و الذي سندل عليه لاحقا في جزء المصادر



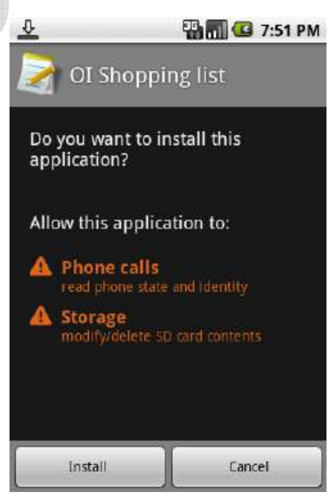
قم بالنقر على زر (Terms & Download) لتظهر لك شاشة اتفاقية الاستخدام



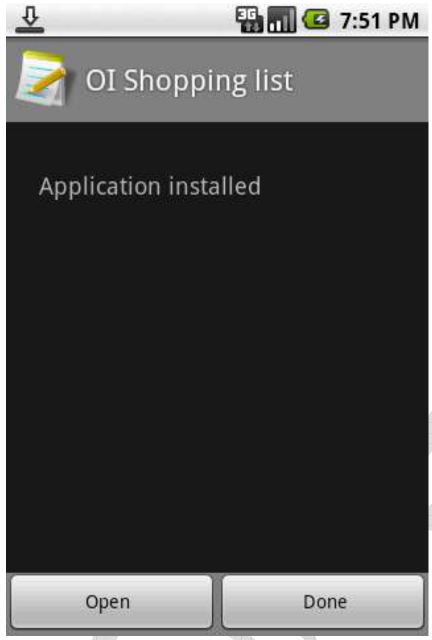
بعد ذلك قم بالنقر على زر (Install) لتبدأ عملية التحميل و التي ستكون في قائمة الإنذارات العلوية (Notifications) إذ يقوم فيها بعملية تحميل البرنامج على جهازك في الخلفية و يمكنك الوصول لها عبر تمرير القائمة العلوية و التي تظهر فيها معلومات البطارية و الساعة



بعد انتهاء البرنامج من التحميل اضغط عليه لتبدأ عملية التنصيب على جهازك



تقوم بالضغط على زر (Install) لتظهر لك شاشة انتهاء التحميل بعد لحظات



قم بالضغط على زر (Open) لتشغيل البرنامج

ملاحظة الملفات التي تقوم بتحميلها يتم وهي من نوع (apk) و يتم الاحتفاظ بها في مجلد (Download) في ذاكرة الجهاز الخارجية (SD Card) و لتستطيع تنصيبها على جهاز أخر دون الحاجة لإعادة الخطوات أو وجودة Slide Me أنت بحاجة لوجود تطبيق أسمة (Apps Installer) و الذي يقوم بالبحث عن الملفات من نوع (apk) على جهازك و تمكينك من تنصيبها مباشرة و يمكنك الحصول على هذا التطبيق من موقع (Slide Me)

قم باستكشاف البرنامج و التعرف على كيفية التعامل مع تطبيق فعلي على أندرويد لتستطيع التعرف على كيفية القيام ببناء واجهات تطبيق مشابهة بنفسك.

تنويه :يمكنك و عبر أداة (DDMS) أن تقوم بأخذ صور عن شاشة المحاكي و ذلك من القائمة (Devices) ثم الذهاب إلى زر الأدوات (شكل صورة) (Screen Capture) و يمكنك أن تقوم بأخذ صورة من المحاكي و حفظها على جهازك كما قمت أنا بذلك عند شرح هذا التدريب

دلع المحاكي

مارأيك لو تقوم بتغيير شكل المحاكي لديك ليصبح بشكل الجهاز ما عليك سوى الذهاب إلى المجلد

android-sdk-windows\platforms\android-1.6\skins\HVGA

و من ثم التلاعب بالصور الموجودة في حال أردت مثلا تغيير صورة الجهاز لتصبح كالتالي

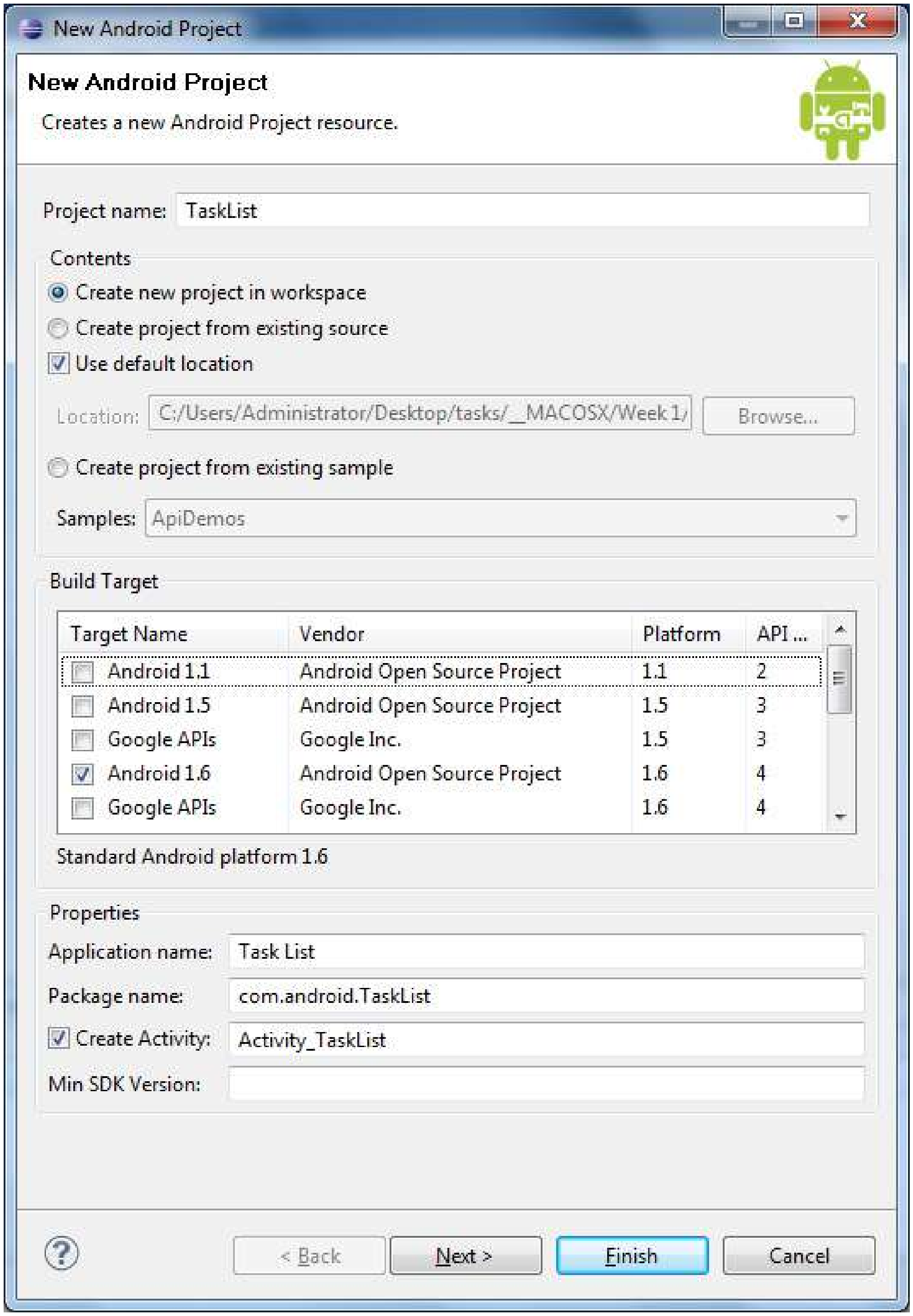


بناء مشروعك الثانى (قائمة المهام)

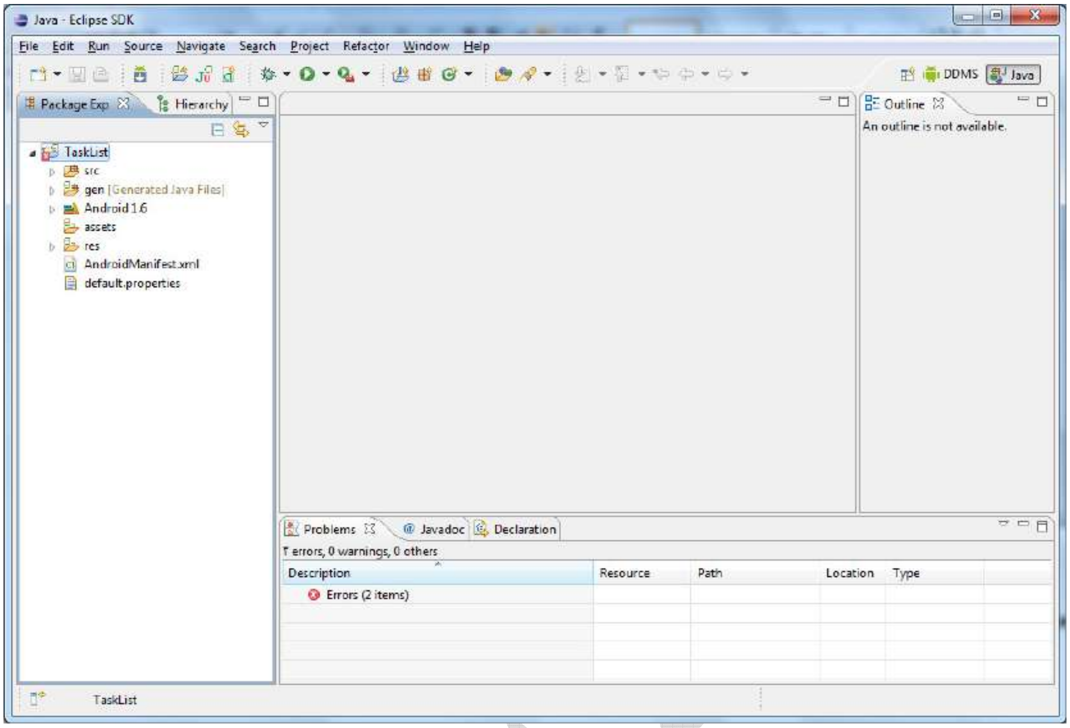
الخطوات الأولى في هذا الكتاب كانت تركز على تعريفك على بيئة التطوير و كيفية تهيئتها و تعليمك أساسيات التعامل معها و لم نقم بكتابة أي سطر كود حتى الآن و لذا فعلينا البدء في التشمير عن ساعدينا و كتابة أول برنامج فعلي يحتوي على بعض الكود و الذي سنشرح بداخله المزيد عن بيئة التطوير و كيفية التعامل معها.

اختياري وقع على برنامج قائمة المهام بالأساس لأنه لا يأتي ضمنيا مع نظام التشغيل كما أنه بسيط في فهم وظيفته و بنفس الوقت فهو يستخدم الكثير من الخصائص التي سنتعلم منها أساسيات البرمجة الموجهة لاندرويد و منها التعامل مع أكثر من شاشة (نشاط) و حفظ المهام و بناء قائمة أوامر و غيرها من المميزات التي تكسبنا المزيد من المهارة.

قم الآن بتشغيل بيئة التطوير (Eclipse) و ابدأ في إعداد مشروع جديد و لنسيمه (Task List) ليكون في الشكل التالي:

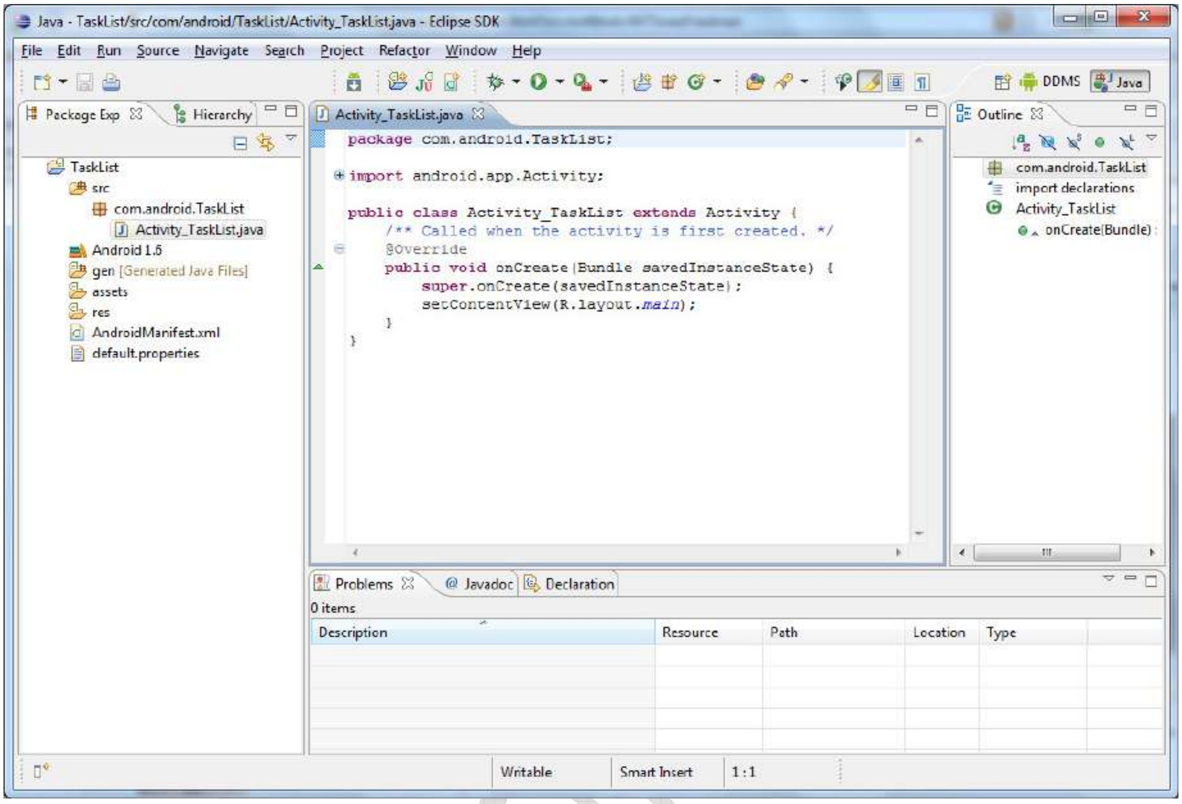


بعد أن تظهر لنا شاشة ملفات المشروع كما في الشكل التالي

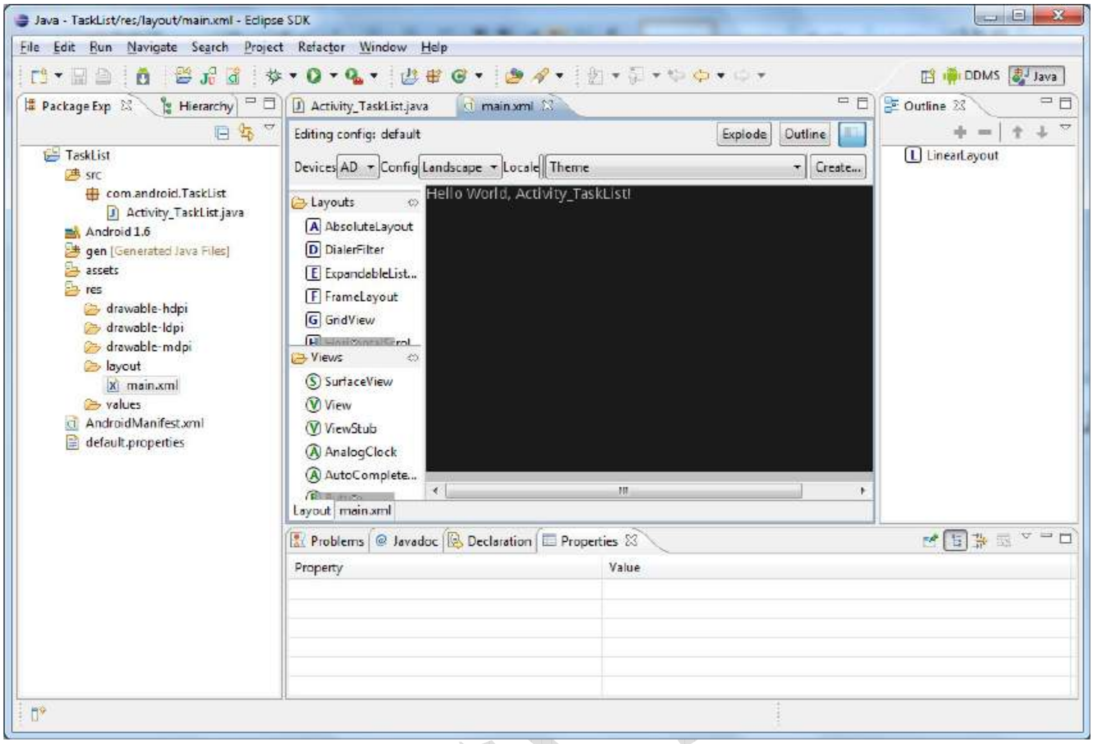


لربما تلاحظ وجود علامة (X) باللون الأحمر على رأس الشجرة و التي تدل على وجود خطاء في المشروع و لتصحيحها قم باستخدام (Project >> Clean) كما ذكرنا سابقا عندما قمنا ببناء أول مشروع و ذلك لتنظيف المشروع من أي أخطاء بسبب المعالج.

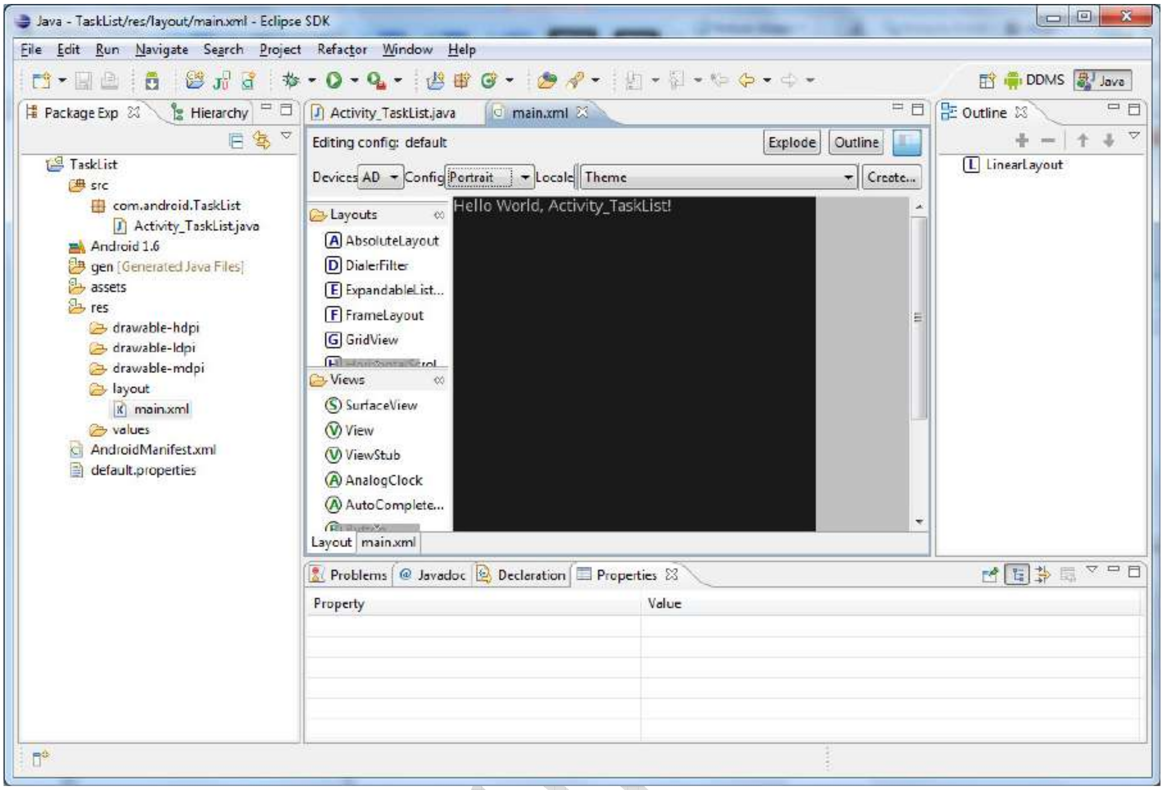
لنستعرض الكود البرمجي و الذي قام المعالج بإنشائه لأول شاشة (نشاط) لنا بشكل افتراضي و لنذهب إلى مجلد (src) و من تحته نذهب إلى الحزمة (com.android.TaskList) ومن ثم من تحتها نقوم بالنقر المضاعف على الملف (Activity_TaskList) ليظهر لنا الكود الأساسي و الذي قام النظام بإضافته و المسئول عن تشغيل الشاشة الرئيسية و التي تحتوي فقط و بشكل افتراضي على مربع عنوان يحتوي أسم المشروع



كما يمكنك استعراض الشاشة (شكل النشاط) الذي قامت بيئة التطوير ببنائه بشكل افتراضي و ذلك تحت مجلد المصادر (res) ثم اختيار (layout) و من تحتها بالنقر المضاعف على الملف (main.xml) ليظهر لدينا شكل الشاشة كما الصورة التالية

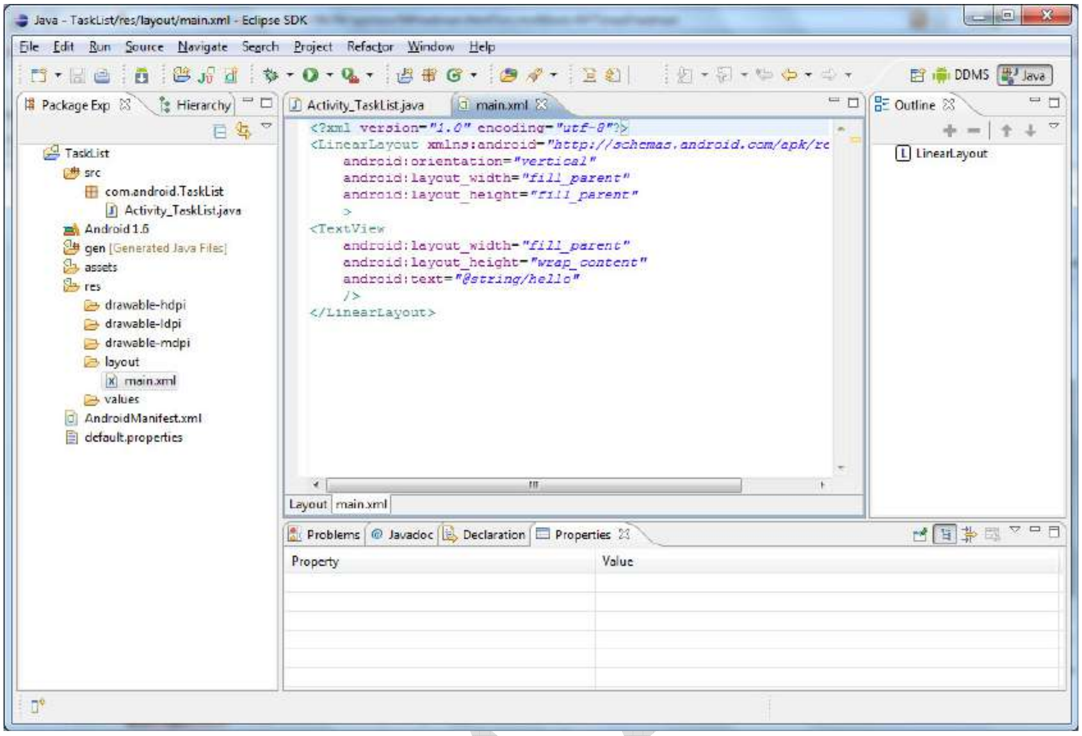


من أعلى لسان التبويب (main.xml) قم بتغيير شكل الشاشة لتصبح بشكل طولي بدلا من شكل عرضي للتحكم أكثر في شكل الشاشة الافتراضية عندما تعمل على الجهاز ، قم بتغيير الخاصية (Config) من القيمة (Landscape) إلى القيمة (Portrait) لتتغير شكل الشاشة قليلا و تصبح بالشكل التالي



من القائمة اليسرى بجانب الشاشة هناك مربعين باسم (Layout) و (Views) و التي يمكنك أن تقوم بسحب عناصرها مباشرة و وضعها على الشاشة و لكننا سنذهب إلى الملف المصدري لنتعرف أكثر على هيكلية الكود و نقوم بعمله بشكل يدوي لنحترف العمل عليه.

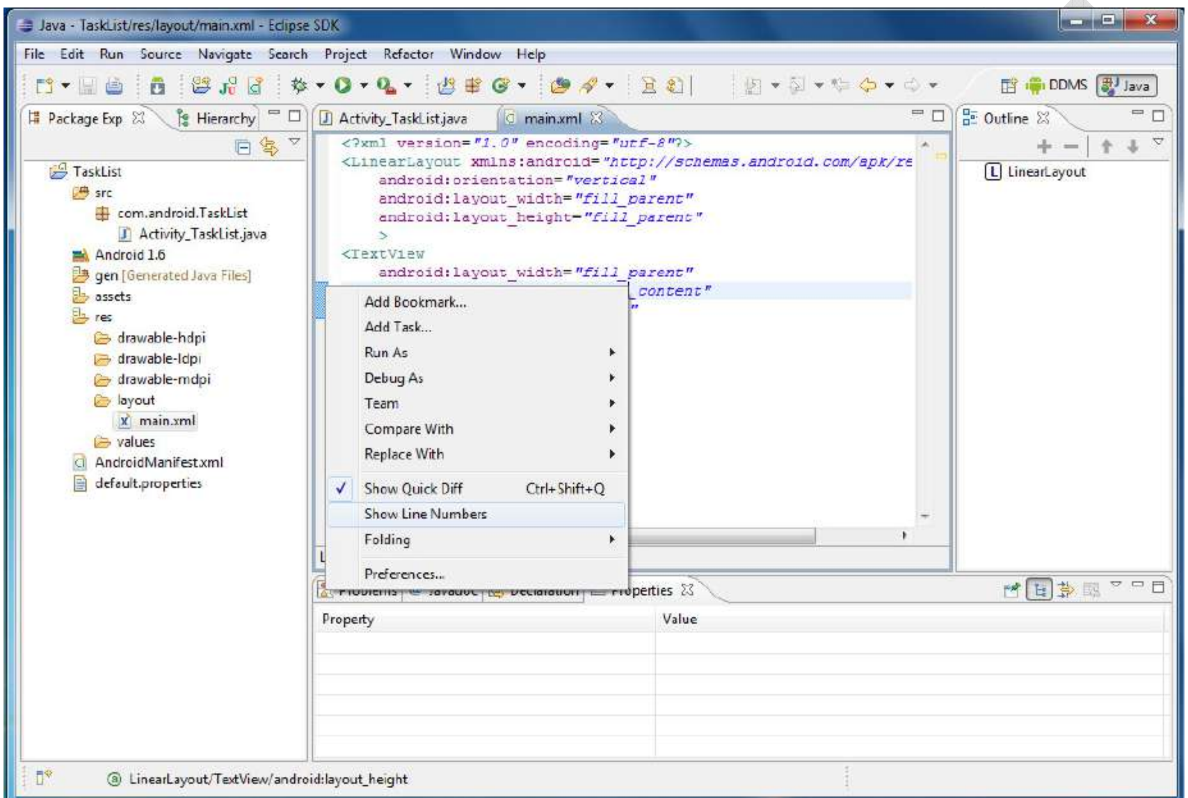
تحت الشاشة في الناحية اليسرى ستجد لسان تبويب (main.xml) و (layout) قم بالتغيير إلى (main.xml) لتظهر لك الشاشة التالية:



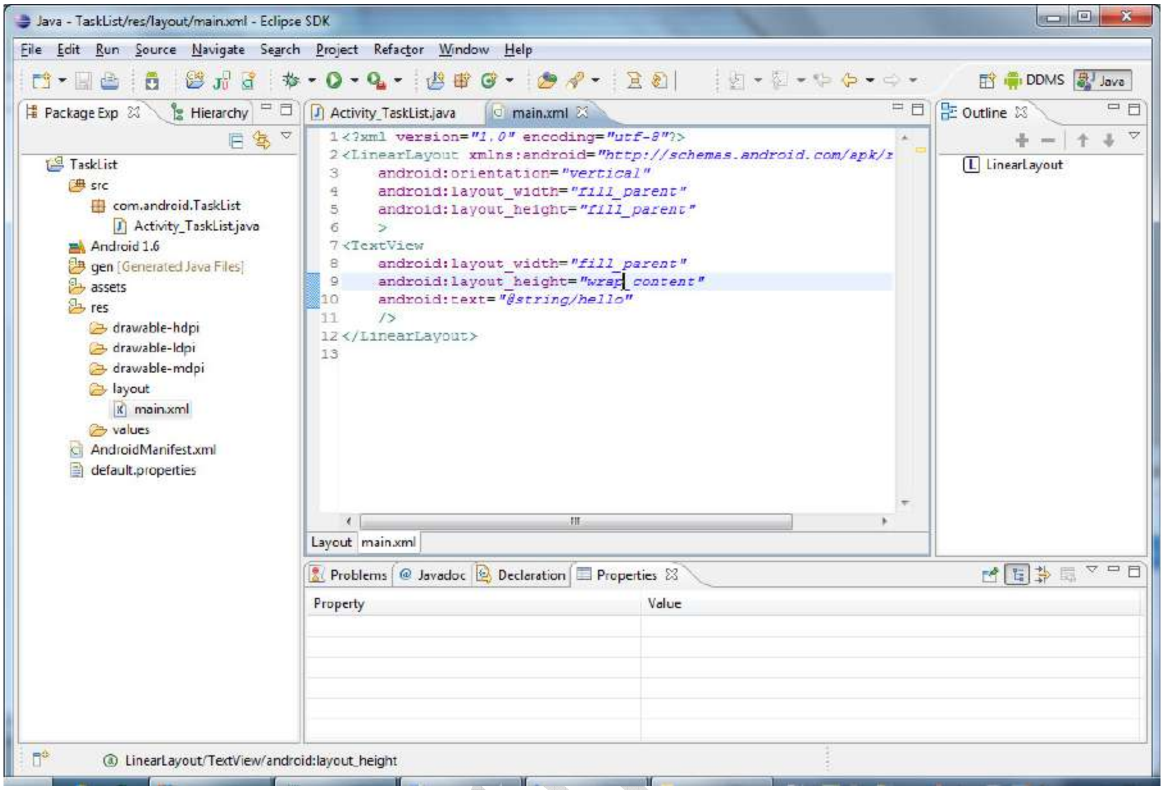
لنتعرف على الأكواد المكتوبة

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />
</LinearLayout>
```

لقد قمت هنا بترقيم الأسطر لسهولة الرجوع لها كما و يمكنك أنت أيضا أن تقوم بإظهار أرقام الأسطر من داخل بيئة التطوير (Eclipse) و ذلك بالضغط بزر الفأرة الأيمن على أقصى اليسار من سطر الكود في المنطقة الفارغة و من ثم من القائمة المنسدلة قم بتحديد خيار (Show Line Numbers)



لتظهر لنا الشاشة كما في الشكل التالي و التي تحتوي أرقام أسطر الكود البرمجي لنستطيع الإشارة إليها لاحقا



الان و بالعودة إلى الكود و شرحه

1 <?xml version=*"1.0"* encoding=*"utf-8"*?>

فتح ترويسة ملف (XML) المعياري و ترميز (UTF-8) الدولي لدعم كل اللغات

2<LinearLayout
xmlns:android=*"http://schemas.android.com/apk/res/android"*
3 android:orientation=*"vertical"*
4 android:layout_width=*"fill_parent"*
5 android:layout_height=*"fill_parent"*>

الأسطر من (2) و حتى (6) هي عبارة عن سطر واحد و لكنه موزع إلى عدة أسطر لسهولة القراءة و يمكنك تغيير ذلك أو الإبقاء على ذلك

المهم في هذا السطر هو اختيار طريقة العرض (LinearLayout) و التي كما ذكرنا في أول الكتاب أحدى طرق العرض من ألٍ (4) طرق عرض مختلفة و التي تقضي برصف العناصر بشكل متتالي من الأعلى إلى الأسفل تباعا

ما بداخل الوسم (LinearLayout) هناك مجموعة من الخصائص التي تحدد طريقة العرض و لعل من أهمها الخاصية (orientation) و التي توصف طريق الرصف المتتالي بشكل عامودي أو أفقي و هي و بشكل افتراضي تقوم بالرصف بشكل عامودي و يمكن تغيرها لتصبح بشكل أفقي إلا بالخاصية (horizontal) و كذلك الحال الخاصيتين (layout_width) و (layout_height) و التي تحدد مكان توضع كل عنصر و التي تأخذ القيمة (*fill_parent*) أي تعبئة المساحة الأب و في هذه الحالة في حال كان الشكل عاموديا كما هو افتراضي فإن كل عنصر سيقوم بتعبئة اسكر من أقصى يسار الشاشة إلى أقصى يمين الشاشة أما في حال كان العرض أفقيا فأنه سيتم التعبئة من الأعلى إلى الأسفل

```
7 <TextView
8    android:layout_width="fill_parent"
9    android:layout_height="wrap_content"
10   android:text="@string/hello"/>
```

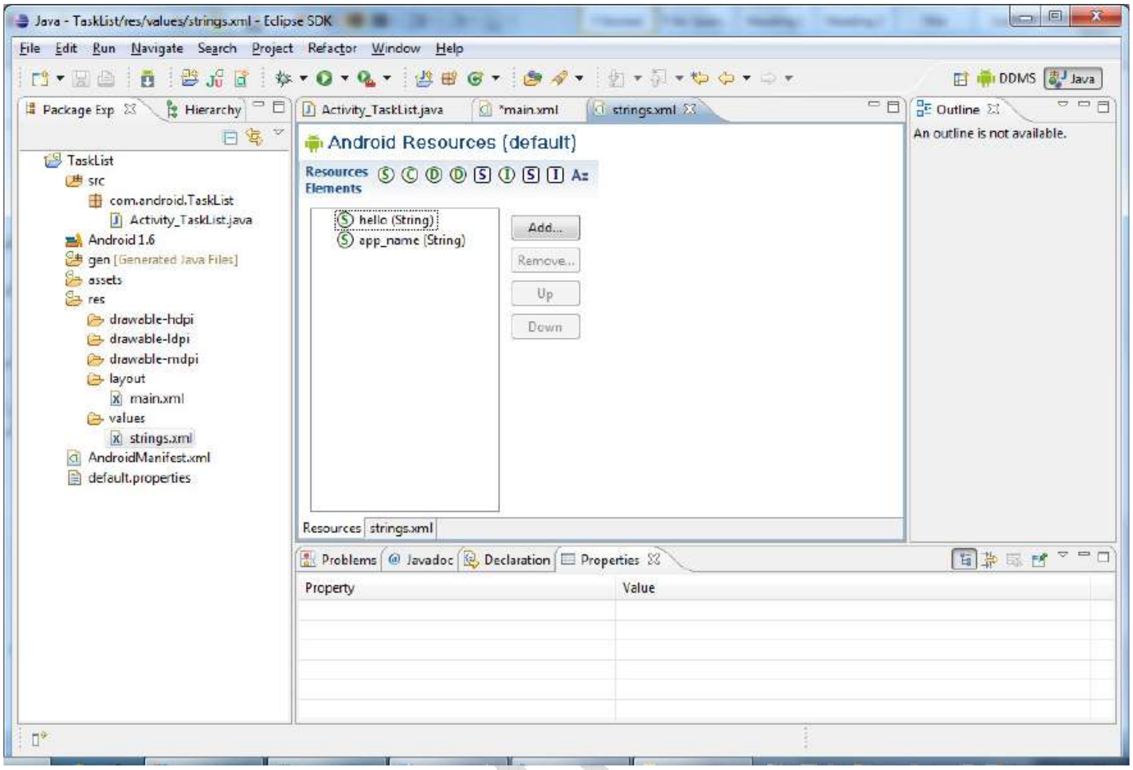
من السطر (7) و حتى السطر (11) هو سطر واحد لتوصيف كائن نصي و التي تم توزيعه على عدة سطور لنفس الاسباب السابقة

هناك بعض الخصائص المشتركة في أغلب العناصر و قد تتغير بعضها بناء على وظيفة العنصر و لكن في حالتنا هذه سنقوم بشرح الخصائص الافتراضية و هي (layout_width) و التي تأخذ القيمة الخاصة بتعبئة الأب (*fill_parent*) و الخاصية (layout_height) و التي تأخذ القيمة (*wrap_content*) و التي تعني أنه يمكن لهذا العنصر التمدد عاموديا و النزول لأكثر من سطر في حال استدعى ذلك بناء على طول النص الداخلي ، أما بالنسبة للخاصية (text) و التي تأخذ قيمتها من ملف المصادر الخارجي و هي القيمة التي ستعرض داخل النص و لا يمكن تغييرها مباشرة من هنا إذ أن (*@string/hello*) هي عنوان القيمة في ملف المصادر و ليست القيمة نفسها و التي سنتعرف عليها بعد شرح هذا الكود مباشرة

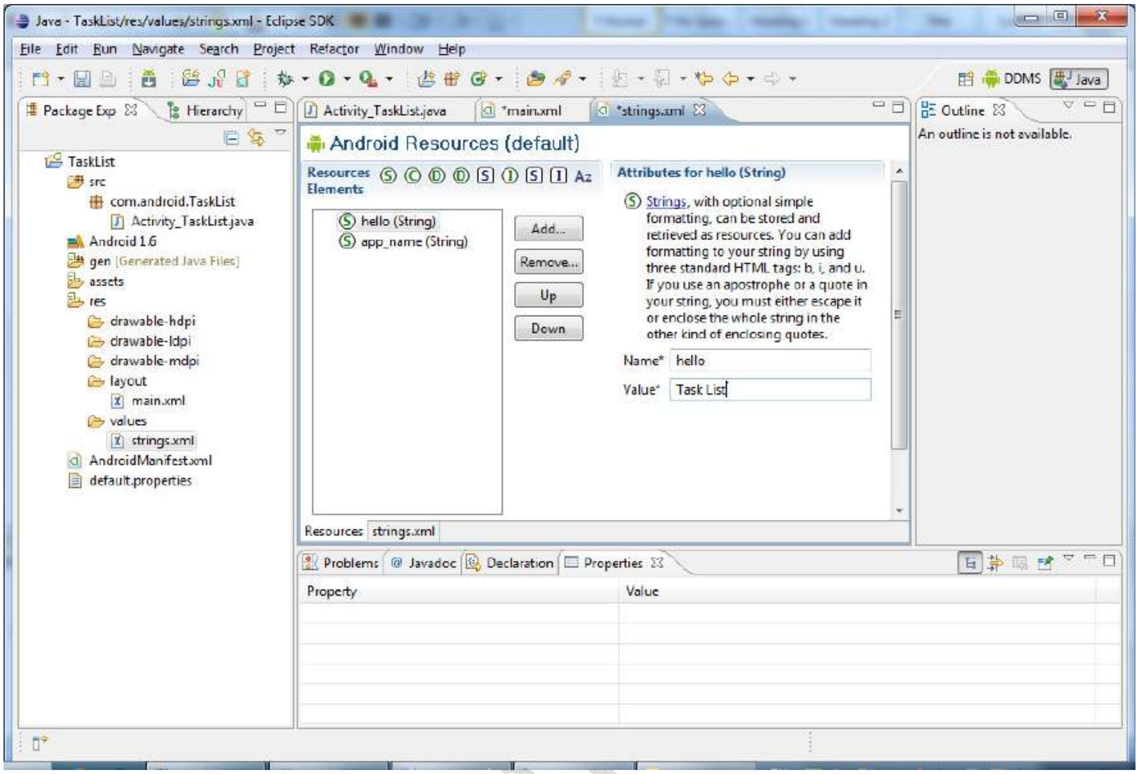
```
12 </LinearLayout>
```

السطر (12) ينهي الوسم (LinearLayout) و الذي ينهي بذلك الكود الخاص بنا

كما ذكرنا سابقا بأن مربع النص (TextView) لديه بعض الخصائص و منها الاسم و التي يأخذ قيمتها من ملف المصادر و الذي يمكن الوصول له من ملفات المشروع تحت المجلد (res) و من ثم مجلد (Value) و بعدها قم بالنقر على اسم الملف () لتظهر الشاشة التالية



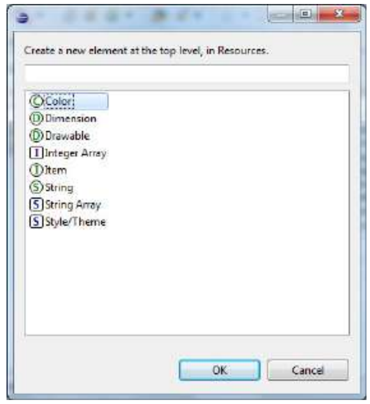
الان يمكننا تعديل قيمة التي بداخل النص من هنا و ذلك بالضغط على اسم المرجع و ليكن مثلا (hell) ومن ثم عندما تفتح النافذة قم بتعديل القيمة من (Hello World, Activity_TaskList!) إلى (Task List) لتصبح بالشكل التالي



الان يمكنك أيضا تغيير المرجع (app_name) و تغيير القيمة فيه من (Task List) إلى (My 1st Application) والتي ستغير عنوان التطبيق في أعلى النافذة و التي ستظهر عندما نقوم بتشغيل البرنامج في المرة القادمة

بعد ذلك سنضيف مرجع أخر عبارة عن لون خلفية النص لنتعرف على كيفية تغيير القيم و استخدامها

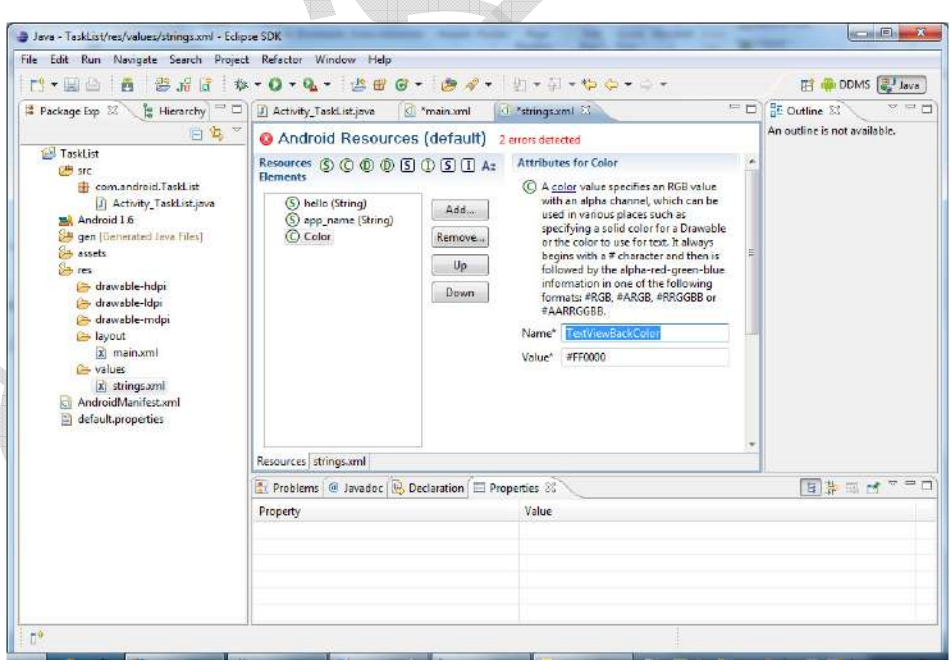
قم بالضغط على زر (add) الموجود بجانب قائمة المراجع لتظهر الشاشة التالية



قم باختيار المرجع (color) من ثم اضغط على زر (OK) ليظهر لدينا عنصر مرجع جديد في القائمة مع تركنا لتعبئة هذا العنصر بالقيم حيث قمنا بتعريف كل من

Name : TextViewBackColor
Value: #FF0000

كما في الشاشة التالية



حيث أن القيمة (#FF0000) تستخدم معادلة ألوان (RGB) و المستخدمة في كود الألوان في (HTML) مثلا و التي تكون فيها عبارة أن أول خانتين (FF) هي للقيمة (R) و التي تعني (Red) و من ثم الخانتين اللاحقتين (00) خاصة بالقيمة (G) و التي تعني (Green) و من ثم الخانتين الأخيرتين (00) خاصة بالقيمة (B) أي (Blue) و بذلك يمكن خلط الألوان الرئيسية الثلاثة لتكوين لون جديد و في حالتنا هذه قم بوضع قيمة للون الأحمر و صفر للونين الأخضر و الأزرق و ذلك للحصول على اللون الأحمر فقط

تحتاج الآن لإغلاق لسان تبويب (string.xml) و ذلك لكي يقوم بحفظ القيم في المشروع و يمكن استخدامها في المشروع نفسه

الان بالعودة إلى الكود الموجود في (main.xml) تحت المجلد (layout) و قمنا بالتعديل على الكود الخاص بمربع النص ليصبح بالشكل التالي

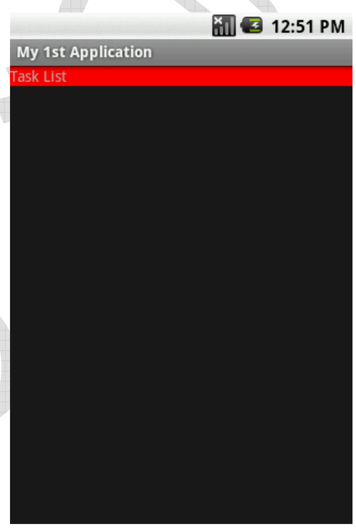
```
<TextView
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:text="@string/hello"
android:background="@color/TextViewBackColor"   />
```

حيث أننا قمنا بإضافة الخاصية (background) و أسندنا لها قيمة اللون التي عرفناها في ملف المصادر (*TextViewBackColor*) الآن بعد هذه التعديلات البسيطة قم بتشغيل برنامجك لترى أثر التعديلات التي قمنا بها
قد يطلب منك البرنامج حفظ المشروع برسالة فقم بالضغط على زر موافق



و لترى بعدها تطبيقنا و هو يعمل على جهاز المحاكي ليكون بالشكل التالي

حيث أن عنوان التطبيق أصبح (My 1st Application) و القيمة النصية في أول سطر أصبحت (Task List) و كذلك ستلاحظ بأن مربع النص أصبح بخلفية حمراء اللون



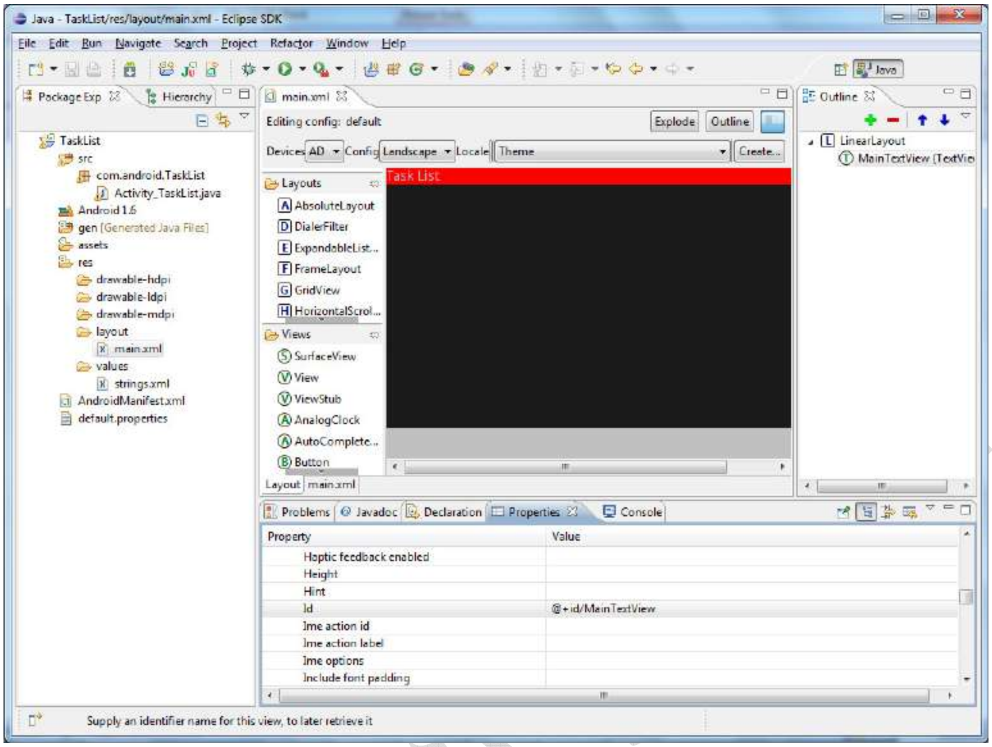
تنويه: كما رأينا كيفية إسناد القيم من ملف (String) و التي تتيح لنا تغيير هذه القيم لاحقا مثلا في حال أتحنا للمستخدم تغيير واجهة التطبيق حيث تقوم بالدلالة على ملف (String) أخر بدلالة اللغة

المحلية مثلا و كذلك الحال يمكننا تمرير النص مباشرة دون الحاجة للدلالة إلى المرجع بالتعويض مباشرة كما الكود الحالي

```
<TextView
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:text="Task List"
   android:background="@color/TextViewBackColor"  />
```

نحن الآن بحاجة إلى خطوة إضافية و هي إعطاء عنصر صندوق النص (TextView) اسم وحيد و ذلك لكي نستطيع التعامل معه عن طريق الكود الخاص بنا و ليكن مثلا (MainTextView) و لنقوم بذلك هناك طريقتين إما عن طريق الكود مباشرة أو عن طريق استخدام صندوق الخصائص و الذي سنتعرف عليه هذه المرة ليقوم هو بالعمل بدلا عنا

قم بالعودة إلى الشاشة تحت مجلد (res >> layout) ليظهر شكل الشاشة الخاصة بالواجهة و في أسفلها ستجد لسان تبويب (Properties) و لتقم بالبحث عن الخاصية (id) و لتقم بإسناد القيمة (MainTextView) إليها ليكون بالشكل التالي



عند انتهائك من هذه النقطة ستقوم بيئة التطوير بإضافة سطر إلى النظام لتعريف هذا الكائن و هو ضمن خصائص مربع النص داخل الملف (main.xml) ليكون بالشكل التالي

```
<TextView
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:text="Task List"
android:background="@color/TextViewBackColor"
   android:id="@+id/MainTextView"/>
```

أو في حال أردت أن تقوم بتعريفه مباشرة من الكود فعليك أن تقوم بكتابته بالشكل التالي

```
android:id="@+id/MainTextView"
```

و التي تقوم بإخبار بيئة التطوير بأننا نقوم بتعريف عنصر جديد باسم (MainTextView) و عندها سنتمكن من التعامل معه دون الحاجة لتعريفه في مكان أخر إلا أن محدودية ذلك تقع أنه لا يمكنك التعامل مع هذا العنصر ضمن الكائنات التي قبله مثله فإنها لن تعرف انه موجود مثلا في حال استخدمنا طريقة العرض (Relative) و أردنا أن تتخاطب هذه الكائنات مع بعضها البعض و تحتاج إلى التعامل معها عن طريق الكائنات ألاحقة

حيث أن (@+id/) و التي وضعناها قبل الاسم تخبر النظام بأن هناك كائن جديد تم تعريفه هنا و لاحقا يمكن التعامل معه مباشرة عبر المرجع مثلا

@id/MainTextView

بدون وجود إشارة الموجب (+) إذ أننا الآن نتعامل معه و ليس تعريفه كما سترى لاحقا

اضافة شاشة جديدة للمشروع (نشاط جديد)

الآن لنقوم بإنشاء شاشة جديدة و هي التي سوف نصل إليها عن طريق قائمة من الشاشة الرئيسية و وظيفة هذه الشاشة هي تمكيننا من كتابة بيانات المهمة و التي تحتوي على مربعي نص أحدهما يحتوي على عنوان المهمة (Title) و الثاني يحتوي على تفاصيل المهمة (Details)

بعد أن انتهينا من بناء الشاشة فلنقم ببناء قائمة (Menu) على الشاشة الرئيسية لتمكننا من استدعاء الشاشة الثانية

بناء قائمة (Menu)

عد الآن إلى الشاشة الأولى (Activity_TaskList) و قم بإضافة سطر استدعاء الدالة (Menu) حتى نستطيع التعامل معها في داخل الكود الخاص بنا و كذلك الحال للدالة الرئيسية (View)

```
import android.view.Menu;
import android.view.View;
```

كما نحتاج إلى إضافة دالة التعامل مع مربعات النص

```
import android.widget.TextView;
```

ليصبح الكود لدينا بهذا الشكل

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.Menu;
import android.widget.TextView;

public class Activity_TaskList extends Activity {
   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

الآن سنضيف بعض الكود لإنشاء (قائمة جديدة)

صناديق الحوار (activities,alertdialog,toast)

يوجد في أندرويد ثلاث أنواع من صناديق الحوار و التي تختلف بالاختلاف الوظيفة ما رأيك الآن لو أضفنا رسالة صغيرة تظهر لفترة عند افتتاح البرنامج تظهر فيها رسالة ترحيب مثلا (Welcome) و يمكنك أن تقوم لاحقا بتضمينها معلومات مفيدة مثلا عدد المهام أو مثلا عدد المهام الواجب أدائها اليوم في حال طورت البرنامج و إضافة خاصية تحديد وقت لإنهاء كل مهمة مثلا الان لاستدعاء دالة صناديق الحوار قم بإضافة الشطر التالي في أعلى الكود

في البداية أنت بحاجة إلى إضافة مرجع

```
import android.widget.Toast;
```

و من ثم قم بإضافة الكود التالي إلى المشروع ليصبح

```
public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);


Toast.makeText(Activity_TaskList.this,"Welcome",0).show()
;
```

```
        TextView[] tvs;
        tvs = new TextView[10];
        for (int i=0; i<10; ++i)
        {
            tvs[i] = new TextView(this);
            tvs[i].setText("hi");
        }
    }
```

الآن عند تشغيل التطبيق ستظهر مربع حوار للترحيب

سنضيف إلى الكود قائمة تظهر عند الضغط على زر(Menu) و التي سنضع بها زر وحيد يحتوي كلمة حفظ (Save) و يظهر الصورة الافتراضية الخاصة بالحفظ من النظام

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);
        MenuItem item1 = menu.add(0, 1, 0, "Save");
    item1.setIcon(android.R.drawable.ic_menu_save );
        return true;
}
```

الآن عند الضغط على هذا العنصر سنقوم بطباعة قيمة النص داخل الكونسول (سجل العمل)

في البداية تحتاج إلى إضافة مرجع إلى

```
import android.util.Log;
```

و من ثم إضافة الكود التالي

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
        TextView MainTextView =
(TextView)findViewById(R.id.MainTextView);
            switch (item.getItemId())
            {
            case 1 :

        Log.v("Task:",MainTextView.getText().toString());
                    return true;
            }
            return false;
    }
```

و بهذا نكون قد أنهينا المدخل إلى هذا المشروع

نشر التطبيق على سوق أندرويد

لنشر التطبيق على سوق أندرويد تحتاج أولا إلى تسجيل حساب خاص بك للبيع و الذي يتطلب دفع مبلغ (50) دولار أمريكي و لمرة واحدة

ثم أنت بحاجة إلى إنشاء مفتاح خاص لكل تطبيق ليكون بمثابة بصمة أو توقيع رقمي لهذا التطبيق ، و في حال إنشاء هذا الملف تحتاج إلى حفظه في مكان أمن ، حيث عند قيامك برفع التطبيق على سوق أندرويد بهذا التوقيع فلن تستطيع أن تقوم برفع أي تحديثات جديدة إلا بنفس التوقيع الذي قمت به بأول مرة

تسجيل مفتاح خاص لتطبيقك

ابحث في جهازك عن أداة (keytool) و الموجودة داخل مجلد (JDK) و ليكن على المسار التالي مثلا

C:\Program **Files\Java\jdk1.6.0_18\bin\keytool.exe**

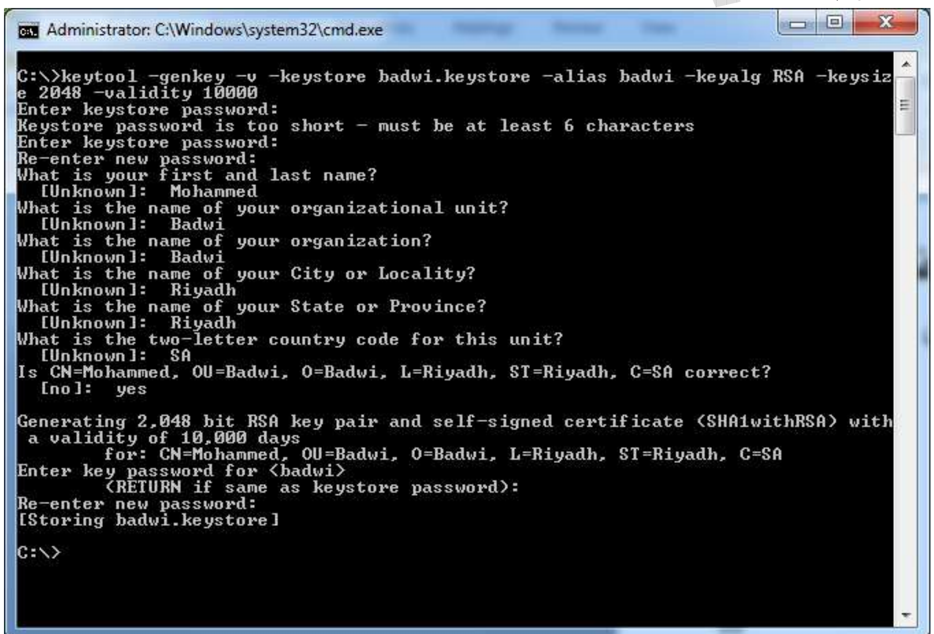
من سطر الأوامر في جهازك قم بكتابة الأمر التالي

keytool -list -alias alias_name -keystore my-release-key.keystore

حيث أنك ستستبدل بعض العوامل بأخرى و ليكن مثلا

keytool -genkey -v -keystore badwi.keystore -alias badwi -keyalg RSA -keysize 2048 -validity 10000

لاحظ اسم التطبيق (badwi) و عندها ستظهر لك بعض المدخلات التي تحتاج إلى تعريفها و ذلك من أجل التشفير الخاص لإنشاء ملف التعريف الخاص



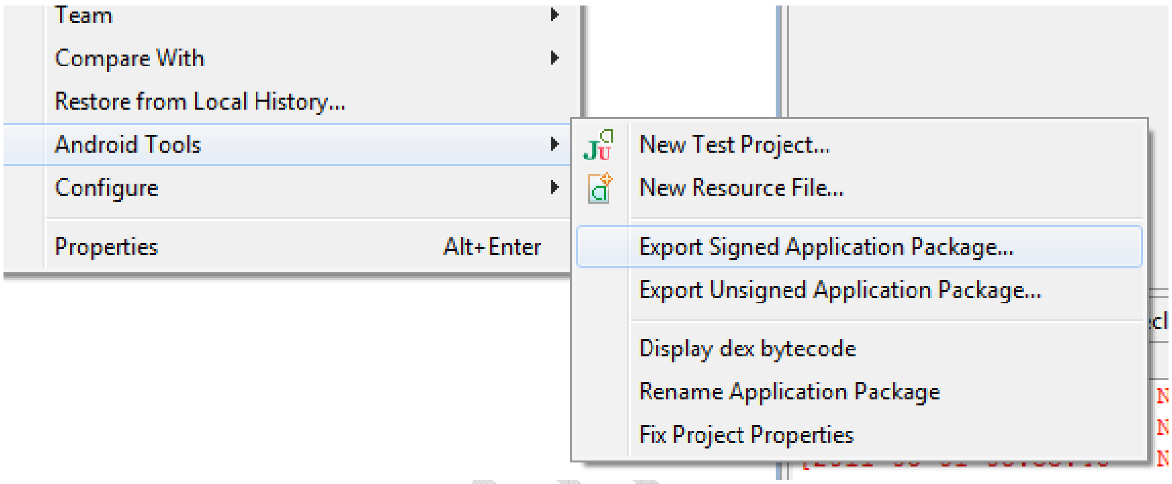
الآن ابحث عن الملف (badwi.keystore) و الذي يكون عادة في المسار:

Windows Vista or 7:
C:\Users\<user>\.android\badwi.keystore

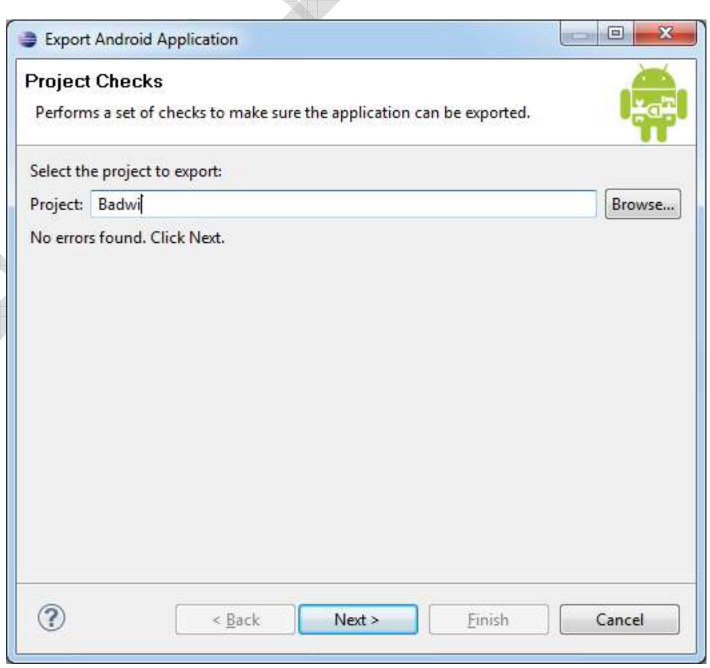
Windows XP:
C:\Documents and Settings\<user>\.android\badwi.keystore

تصدير التطبيق مع المفتاح

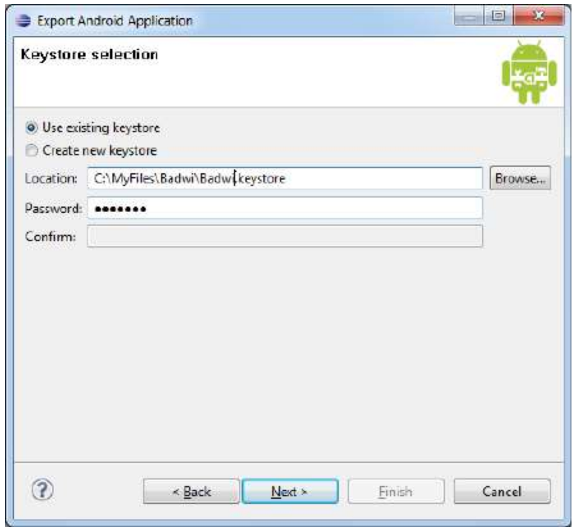
بعد هذه الخطوة تحتاج لربط هذا الملف مع الملف المنتج و ذلك عن طريق بيئة التطوير ،اذهب إلى اسم المشروع و اضغط بزر الفأرة الأيمن



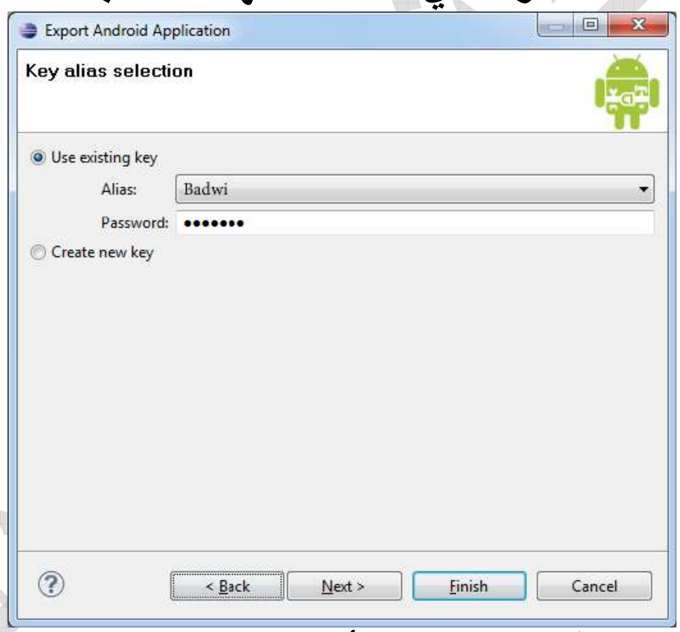
ليظهر لك مربع معالج تصدير الملف و الذي يحوي اسم المشروع



اختر التالي



في هذه الشاشة تحتاج إلى تحديد مكان ملف المفتاح الذي انشئناها سابقا و كذلك كلمة السر الذي استخدمتها أثناء بناء المفتاح



اكتب كلمة السر الذي اخترتها أثناء تعريف معرف المفتاح أثناء الإنشاء

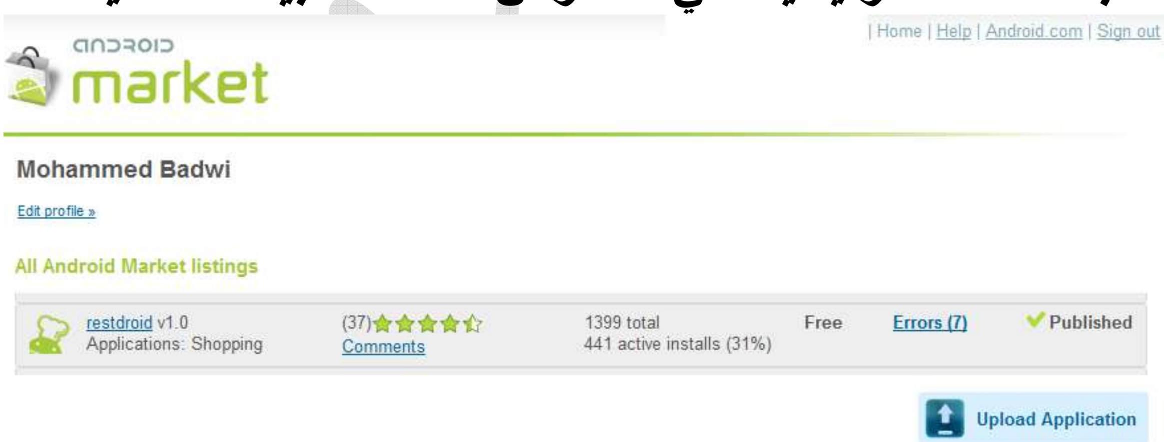


في هذه الشاشة ستحدد أين سوف تقوم بتصدير الملف النهائي و الذي سوف تقوم برفعه على سوق أندرويد

رفع التطبيق على سوق أندرويد

اذهب إلى العنوان التالي
http://market.android.com/publish/Home

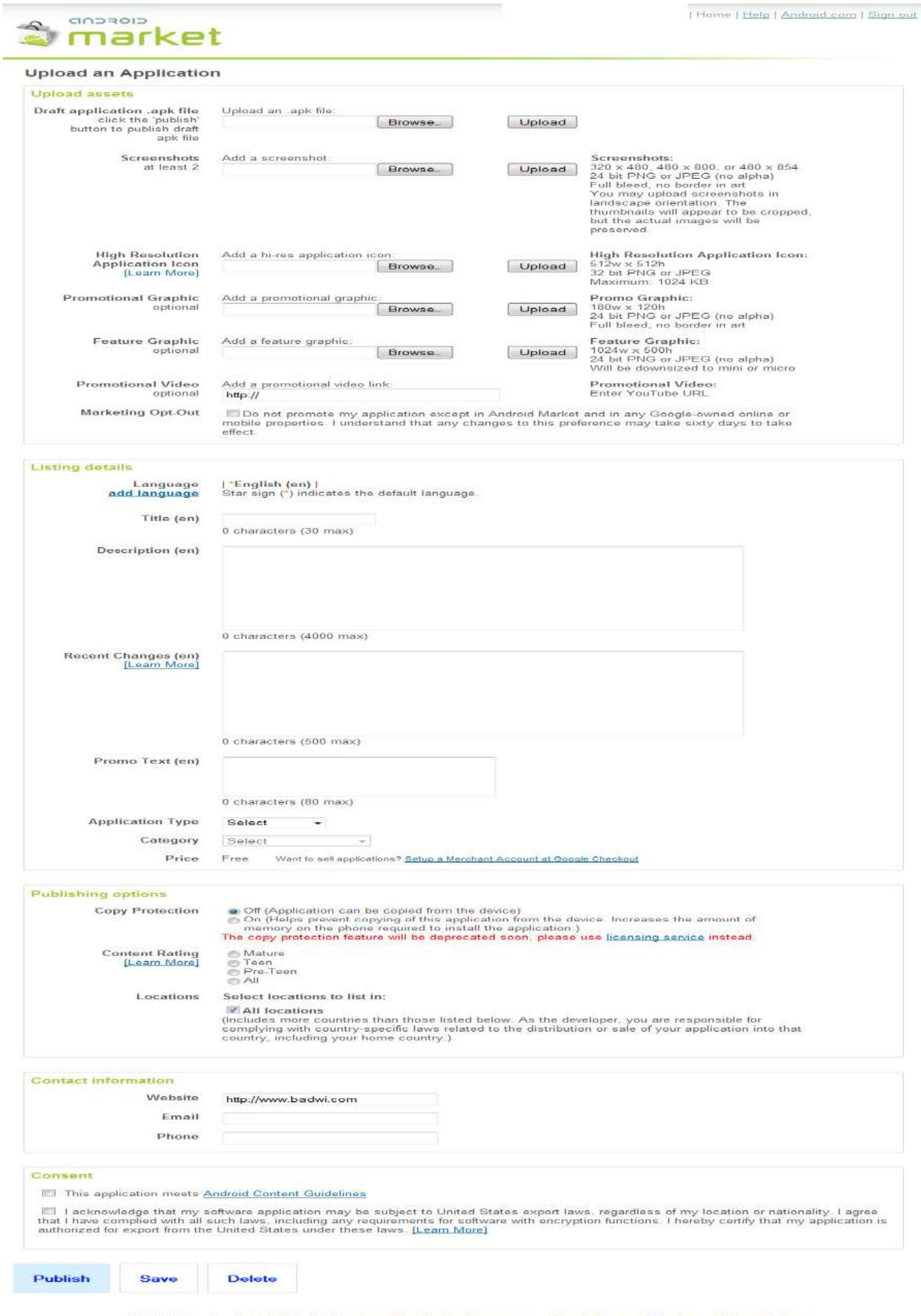
ستجد الصفحة الرئيسية التي تستعرض قائمة التطبيقات الحالية



اضغط على زر (Upload Application)

لتظهر لك شاشة إضافة تطبيق جديد ، ما عليك سوى رفع التطبيق النهائي الذي قمنا بتصديره و الصور الخاصة بالتطبيق و الأيقونات و المعلومات الخاصة بك

الشاشة الخاصة بالإضافة في الصفحة التالية، و في الصفحة التي تليها توجد صورة للشاشة بعد رفع البيانات


android market
| Home | Help | Android.com | Sign out
Upload an Application
Upload assets
Draft application .apk file
click the 'publish' button to publish draft apk file
Upload an .apk file:
Browse...
Upload
Screenshots
at least 2
Add a screenshot:
Browse...
Upload
Screenshots:
320 x 480, 480 x 800, or 480 x 854
24 bit PNG or JPEG (no alpha)
Full bleed, no border in art
You may upload screenshots in landscape orientation. The thumbnails will appear to be cropped, but the actual images will be preserved.
High Resolution Application Icon
[Learn More]
Add a hi-res application icon:
Browse...
Upload
High Resolution Application Icon:
512w x 512h
32 bit PNG or JPEG
Maximum: 1024 KB
Promotional Graphic
optional
Add a promotional graphic:
Browse...
Upload
Promo Graphic:
180w x 120h
24 bit PNG or JPEG (no alpha)
Full bleed, no border in art
Feature Graphic
optional
Add a feature graphic:
Browse...
Upload
Feature Graphic:
1024w x 500h
24 bit PNG or JPEG (no alpha)
Will be downsized to mini or micro
Promotional Video
optional
Add a promotional video link:
http://
Promotional Video:
Enter YouTube URL
Marketing Opt-Out
Do not promote my application except in Android Market and in any Google-owned online or mobile properties. I understand that any changes to this preference may take sixty days to take effect.
Listing details
Language
add language
| *English (en) |
Star sign (*) indicates the default language.
Title (en)
0 characters (30 max)
Description (en)
0 characters (4000 max)
Recent Changes (en)
[Learn More]
0 characters (500 max)
Promo Text (en)
0 characters (80 max)
Application Type
Select
Category
Select
Price
Free
Want to sell applications? Setup a Merchant Account at Google Checkout
Publishing options
Copy Protection
Off (Application can be copied from the device)
On (Helps prevent copying of this application from the device. Increases the amount of memory on the phone required to install the application.)
The copy protection feature will be deprecated soon, please use licensing service instead.
Content Rating
[Learn More]
Mature
Teen
Pre-Teen
All
Locations
Select locations to list in:
All locations
(Includes more countries than those listed below. As the developer, you are responsible for complying with country-specific laws related to the distribution or sale of your application into that country, including your home country.)
Contact information
Website
http://www.badwi.com
Email
Phone
Consent
This application meets Android Content Guidelines
I acknowledge that my software application may be subject to United States export laws, regardless of my location or nationality. I agree that I have complied with all such laws, including any requirements for software with encryption functions. I hereby certify that my application is authorized for export from the United States under these laws. [Learn More]
Publish
Save
Delete
© 2011 Google - Android Market Developer Distribution Agreement - Google Terms of Service - Privacy Policy


android market
Upload an Application
Upload assets
Application .apk file
restdroid.devdroid.com (347k) Published
RestDroid
VersionName: 1.0
VersionCode: 1
[Upload Upgrade]
Localized to: default
This apk requests 2 permissions that users will be warned about
This apk requests 3 features that will be used for Android Market filtering
Screenshots
at least 2
add another
Screenshots:
320 x 480, 480 x 800, or 480 x 854
24 bit PNG or JPEG (no alpha)
Full bleed, no border in art
You may upload screenshots in landscape orientation. The thumbnails will appear to be cropped, but the actual images will be preserved.
RESTDROID
جوعان
Replace this image | delete
High Resolution Application Icon
[Learn More]
Add a hi-res application icon:
Browse...
Upload
High Resolution Application Icon:
512w x 512h
32 bit PNG or JPEG
Maximum: 1024 KB
Promotional Graphic
optional
Promo Graphic:
180w x 120h
24 bit PNG or JPEG (no alpha)
Full bleed, no border in art
Replace this image | delete
Feature Graphic
optional
Add a feature graphic:
Browse...
Upload
Feature Graphic:
1024w x 500h
24 bit PNG or JPEG (no alpha)
Will be downsized to mini or micro
Promotional Video
optional
Add a promotional video link:
http://
Promotional Video:
Enter YouTube URL
Marketing Opt-Out
Do not promote my application except in Android Market and in any Google-owned online or mobile properties. I understand that any changes to this preference may take sixty days to take effect.
Listing details
Language
add language
| *English (en) |
Star sign (*) indicates the default language.
Title (en)
restdroid
9 characters (30 max)
Description (en)
restdroid will help you in finding best restaurants in your area.
ريستدرويد يساعدك على ايجاد أفضل المطاعم في المنطقة القريبة منك
127 characters (4000 max)
Recent Changes (en)
VersionName: 1.0
[Learn More]
0 characters (500 max)
Promo Text (en)
restdroid will help you in finding best restaurants in your area.
65 characters (80 max)
Application Type
Applications
Category
Shopping
Price
Free
Publishing options
Copy Protection
Off (Application can be copied from the device)
On (Helps prevent copying of this application from the device. Increases the amount of memory on the phone required to install the application.)
The copy protection feature will be deprecated soon, please use licensing service instead.
Content Rating
[Learn More]
Mature
Teen
Pre-Teen
All This rating option has been disabled by the Android Market team
Locations
Select locations to list in:
All locations
(Includes more countries than those listed below. As the developer, you are responsible for complying with country-specific laws related to the distribution or sale of your application into that country, including your home country.)
Contact information
Website
http://www.devdroid.org
Email
Phone
+966503415343
Consent
This application meets Android Content Guidelines
I acknowledge that my software application may be subject to United States export laws, regardless of my location or nationality. I agree that I complied with all such laws, including any requirements for software with encryption functions. I hereby certify that my application is authorized for from the United States under these laws. [Learn More]
Unpublish
Save
© 2011 Google - Android Market Developer Distribution Agreement - Google Terms of Service - Privacy Policy



تطبيق RestDroid-mini

تطبيقRestDroid-mini

يتناول هذا الفصل شرح الكود البرمجي الخاص بتطبيق RestDroid والذي طرحناه كأول مشروع من مشاريع مجتمع مطوري أندرويد العرب بالتعاون مع موقع أردرويد المتخصص بكل ماله علاقة بنظام الأندرويد وهواتفه. التطبيق تم طرحه في سوق تطبيقات الأندرويد "أندرويد ماركت وسوق سلايد مي" وكنا قد كتبنا عنه وعن المميزات التي يملكها. وكما ذكرنا من قبل، التطبيق سيكون مفتوح المصدر بشكل كامل ليتمكن المهتمون العرب ببرمجة الأندرويد من دراسة وفهم الشفرة المصدرية والانطلاق نحو مجال تطبيقات الأندرويد. ولكن سنقوم في هذا الكتاب بتقديم شرح كامل لنسخة مصغرة من تطبيق RestDroid الأصلي سميناها ب.RestDroid-mini

النسخة الكاملة من التطبيق تحتوي على الخصائص التالية:

تطبيق RestDroid يستخدم الواجهة البرمجية التطويرية API الخاصة بموقع قيم ليبني عليها المزايا التالية:
-اختيار الدولة والمدينة والبحث عن المطاعم فيها.
-اختيار الدولة والمدينة وعرض أفضل ٥٠ مطعما فيها.
-اختيار الدولة والمدينة والبحث عن مطاعم باستخدام مواصفات معينة (مطعم ايطالي، صيني، نباتي، أسماك، .(...
-عرض المطاعم التي تناسب معلومات البحث وبجانبها التقييمات الخاصة بها.
-عرض المطاعم على خرائط غوغل.
-إمكانية استعراض معلومات الفروع (وقد لا يظهر بعضها على الخريطة وذلك بسبب عدم تحديد مواقعها في الأصل في موقع قيم من قبل المستخدمين.(
-إمكانية استعراض تعليقات المستخدمين على المطاعم.

التطبيق يستخدم التقنيات التالية:
-الاتصال بالانترنت واستخدام ال.API
-خرائط غوغل.

للاستزادة:

مقطع تعريفي على اليوتيوب:
http://www.youtube.com/watch?v=BEDuS6gq4JA

بالنسبة للنسخة المصغرة والتي سنشرحها هنا، سيكتفي التطبيق بعرض زر واحد بالضغط عليه يتم جلب أفضل ٥٠ مطعما في مدينة الرياض-المملكة العربية السعودية ويعرض تقييمها. بالضغط على أي مطعم، سيتم عرض موقعه على خريطة غوغل.

واجهات التطبيق:

الخطوة التي تسبق البرمجة، هي تحديد شكل وعدد الواجهات المطلوب إظهارها في التطبيق. نحتاج في تطبيقنا الى ثلاثة واجهات وهي:

واجهة البداية:

وفيها يظهر شعار التطبيق وزر مكتوب عليه "جلب أفضل ٥٠ مطعما في الرياض."

واجهة المطاعم:

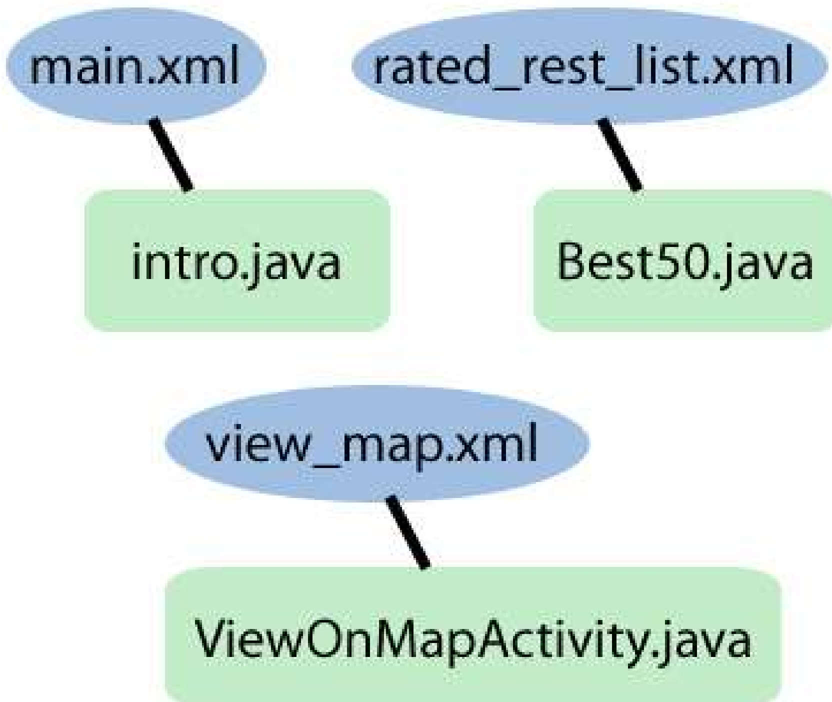
وفيها تظهر أسماء المطاعم وتقييماتها. الشكل المرغوب هو أن يظهر تقييم كل مطعم (العلامة الموزونة) على شكل ٥ نجوم + العلامة الموزونة رقما + عدد الأصوات رقما (العلامة الموزونة هي المعيار الأنسب لمقارنة المطاعم بحسب موقع قيم).

واجهة الخريطة:

وهي الواجهة التي يظهر فيها اسم المطعم ومكانه على خريطة غوغل بحيث يتم تمثيل مكان المطعم بنقطة خضراء.

بعد تصميم الواجهات، نأتي الى الآن الى تصميم المجموعات Classes والدوال Methodsالخاصة بالتطبيق. سنقوم جاهدين بإتباع أفضل السبل في تصميم المجموعات بشكل يسهل على القارىء فهمها وفهم وظيفتها (**نفترض في شرحنا إلمام القارىء بلغة الجافا -مستوى مبتدئ متقدم- وتمكنه من فهم الأكواد التي عرضناها في الصفحات السابقة من هذا الكتاب**)

بداية كما نعرف فإن كل واجهة يقترن بها **مجموعة خاصة** بها إضافة إلى **ملف XML** يحتوي على العناصر المرئية المطلوب إظهارها (الشعار + زر في الواجهة الأولى مثلا) وهذا يعني أننا بحاجة إلى ٣ مجموعات جافا و٣ ملفات XML بالشكل التالي:



طبعا الواجهة الأولى هي intro والثانية هي Best50 والثالثة هي ViewOnMapActivity.

بالإضافة الى الثلاث مجموعات أعلاه، يوجد لدينا عدد من المجموعات المساعدة وهي على النحو التالي:

RestDroidApplication.java

وهي المجموعة الأم التي تعتبر باقي المجموعات أطفالا لها "كونها تقوم بوراثة مجموعة Application والتي تندرج تحتها حكما الأنشطة Activities والمجموعات Classes الموجودة في التطبيق". الهدف من تواجدها هو وضع

الدوال المشتركة بين المجموعات فيها بحيث لا يتكرر الكود البرمجي للدوال في أكثر من مجموعة. بالإضافة الى استخدامها كمستودع للمتغيرات العامة والتي يمكن لأي مجموعة الدخول اليها) Global Variables هذه المتغيرات سيتم توظيفها كوسيلة لتمرير القيم التي تطلبها مجموعة\واجهة من المجموعة التي قبلها). بمعنى آخر أننا لسنا بحاجة الى اللجوء لهذا النوع من المجموعات في كل تطبيق نبرمجه، ولكن في حالة وجود عدد من الدوال المشتركة بين المجموعات فإنه يفضل وضعها في مكان واحد ليسهل فيما بعد التعديل على هذه الدوال.

RatedItem.java

وهي مجموعة بسيطة لكائن Object يمثل المطعم. هذا المطعم لديه الخصائص التالية: رقم مميز + اسم + عدد الأصوات + العلامة الموزونة.

RatedRestaurantListItem.java

ويقترن بها

rest_list_item.xml

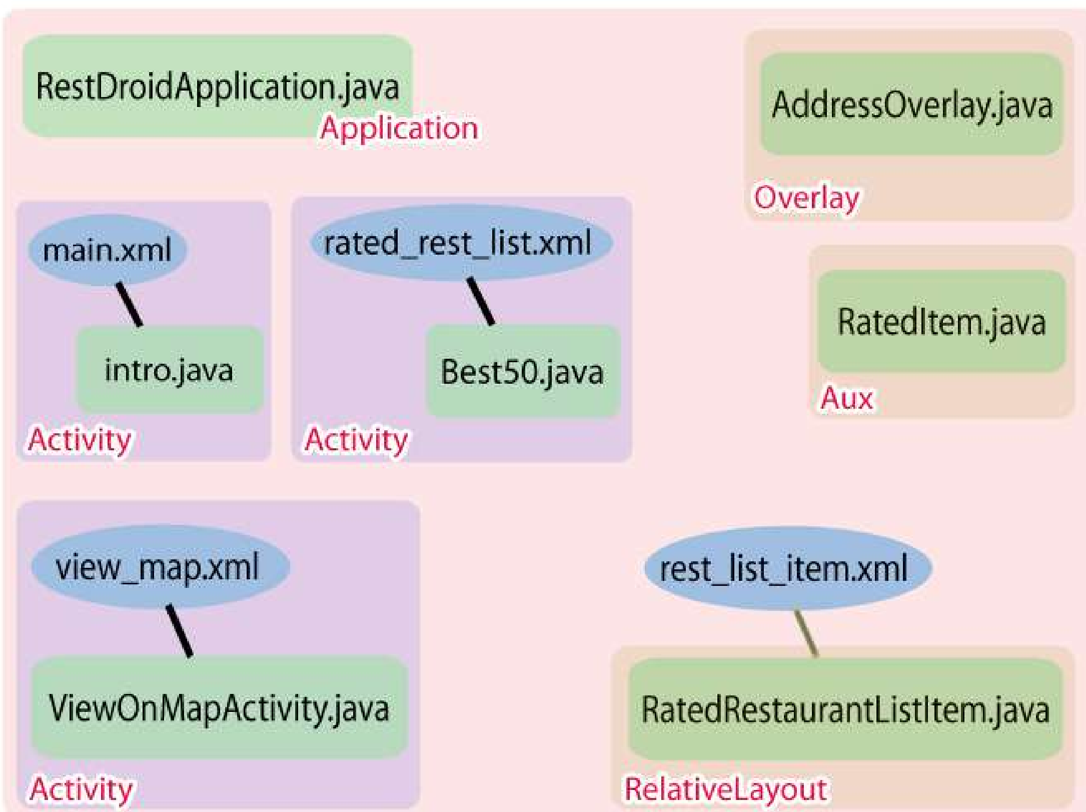
ودور هذه المجموعة وملف الاكس ام ال هو تكوين قالب العرض الخاص بكل مطعم. كما نعرف فإن ما يتم وضعه في ملف الاكس ام ال يظهر في الواجهة عند تشغيلها، فإذا وضعنا به ٣ أزرار وصورة، ستظهر كما هي. **السؤال** هو ماذا لو كنا نريد أن تتغير الواجهة وتتشكل بشكل تفاعلي يتناسب في حالتنا مع عدد المطاعم (في هذا التطبيق لدينا رقم ثابت وهو ٥٠ مطعما، ولكن في التطبيق الأشمل، قد لا تملك احدى المدن أكثر من ٣٠ مطعما) أضف الى أننا لا نريد تكرار عناصر الواجهة ٥٠ مرة؟
الحل هو بتشكيل الواجهة الخاصة بعنصر واحد (أي مطعم واحد) والتحكم برمجيا بعدد مرات التكرار ومحتوى كل عنصر (وهنا تأتي الحاجة لمجموعة RatedItem.javaوالتي سنستقي منها معلومات كل عنصر).



الشكل ١

AddressOverlay.java
هذه المجموعة يتم استخدامها كطبقة تظهر فوق الخريطة بحيث يمكننا رسم نقطة خضراء في المكان المقابل لموقع المطعم.

إذ يصبح لدينا الشكل النهائي للمجموعات كما هو في الشكل ٢.



الشكل ٢

نلاحظ من الشكل أعلاه ما يلي:

-مجموعة RestDroidApplication.java أصبحت مجموعة عامة تستطيع باقي المجموعات التواصل معها كون الأولى ورثت خصائص مجموعة Applicationوهي المجموعة التي تمثل جسد تطبيق الأندرويد (أي أن التطبيق Applicationمكون من مجموعة نشاط Activity واحدة أو أكثر).

-المجموعات الثلاث التي تستمد خصائصها من Activity يجمعها أنها تستطيع رؤية دوال ومتغيرات الجسد الذي يحتويها، أي المجموعة السابقة.

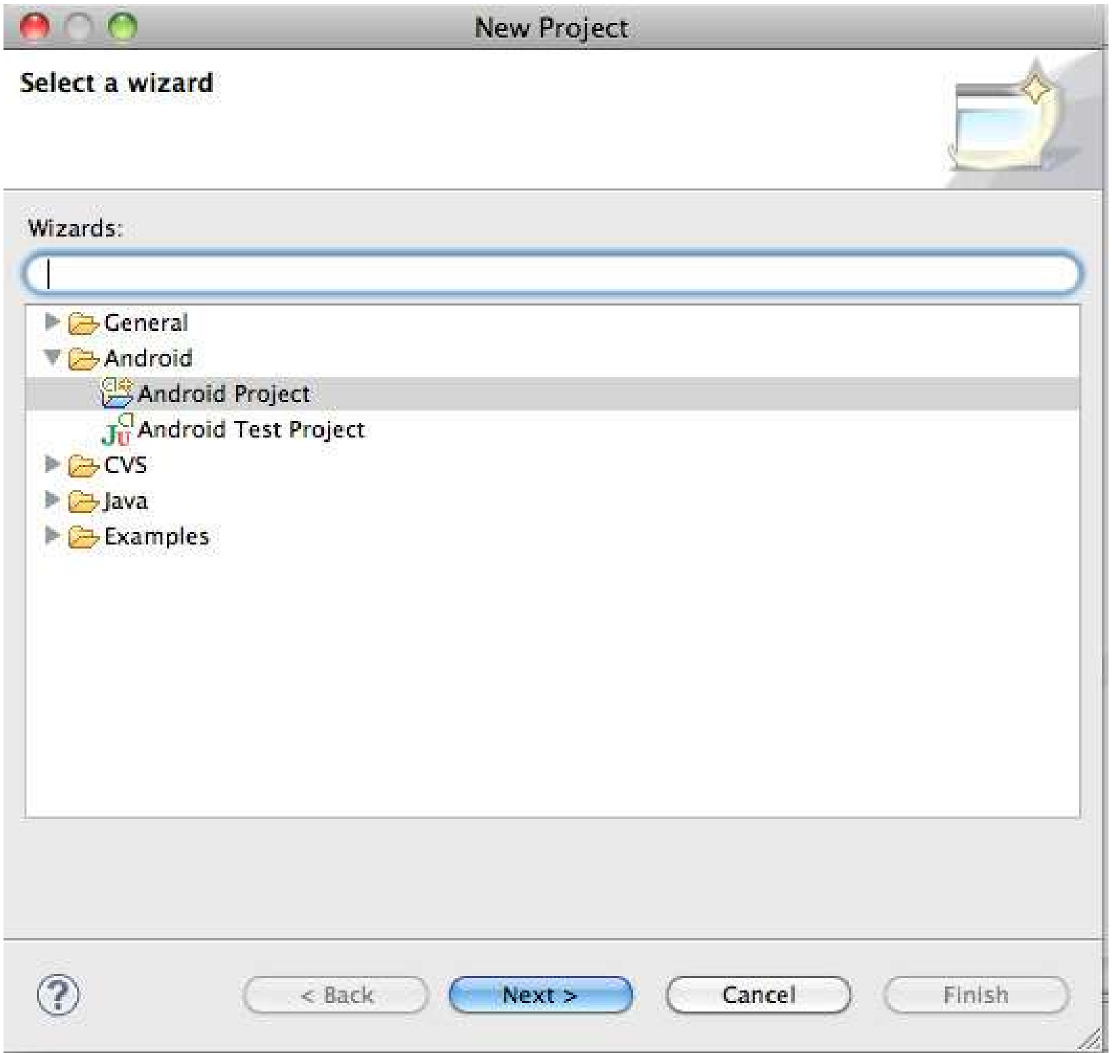
-المجموعات الباقية هي كما ذكرنا مساعد (Auxiliary) وبعضها يستفيد من خصائص مجموعات أعلى ويعيد صياغتها وتعريفها (كمجموعتي RatedRestaurantListItem.javaو AddressOverlay.java) وهو ما سيتم توضيحه لاحقا.

-مجموعة RatedRestaurantListItem.java ترتبط بملف rest_list_item.xmlبشكل مغاير نوعا ما لارتباط باقي المجموعات بملفات الاكس ام ال خاصتها وهذا سيبرز جليا في الشرح القادم (لذلك تم وصل المجموعة وملفها بخط رمادي وليس أسود).

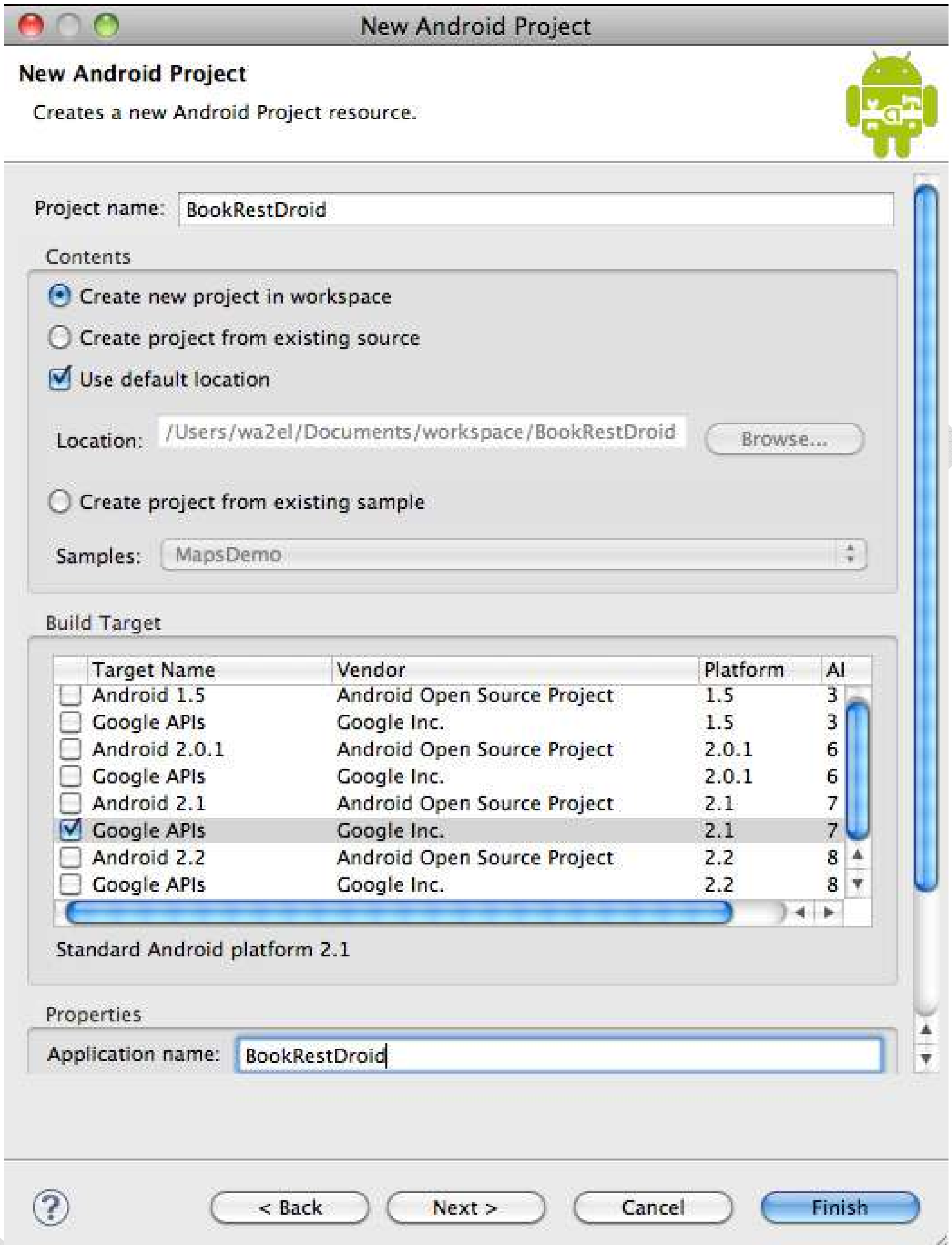
إنشاء التطبيق:

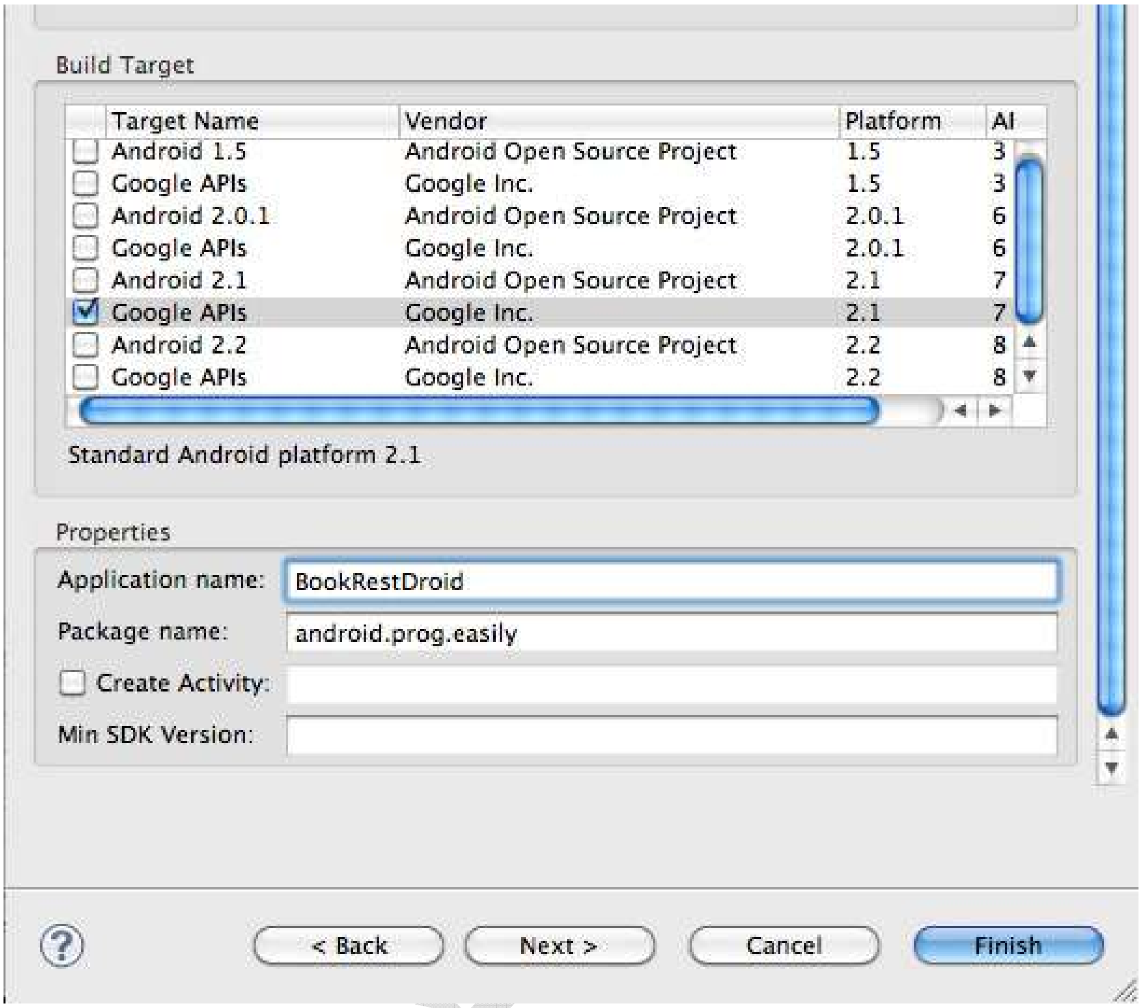
نقوم بتشغيل الاكليبس والتوجه إلى قائمةFile>new>project

نقوم بفتح مجلد Android ونختار منهAndroid Project

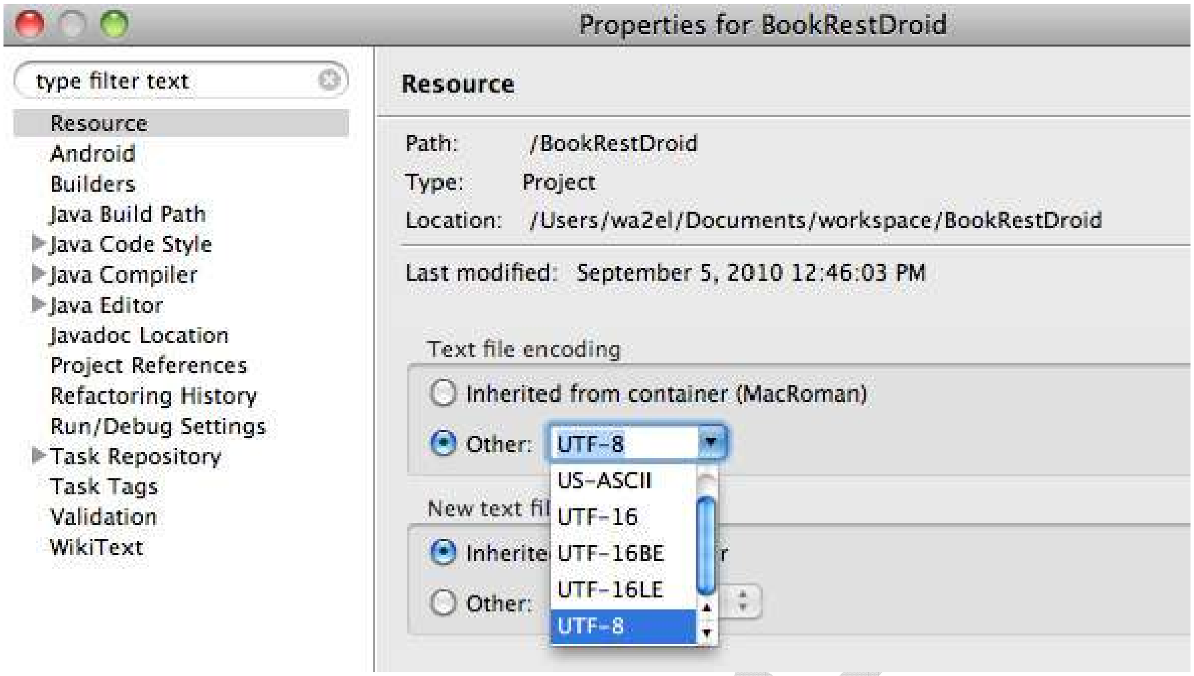


ثم نقوم بمليء البيانات بشكل مشابه لما في الصور التالية:

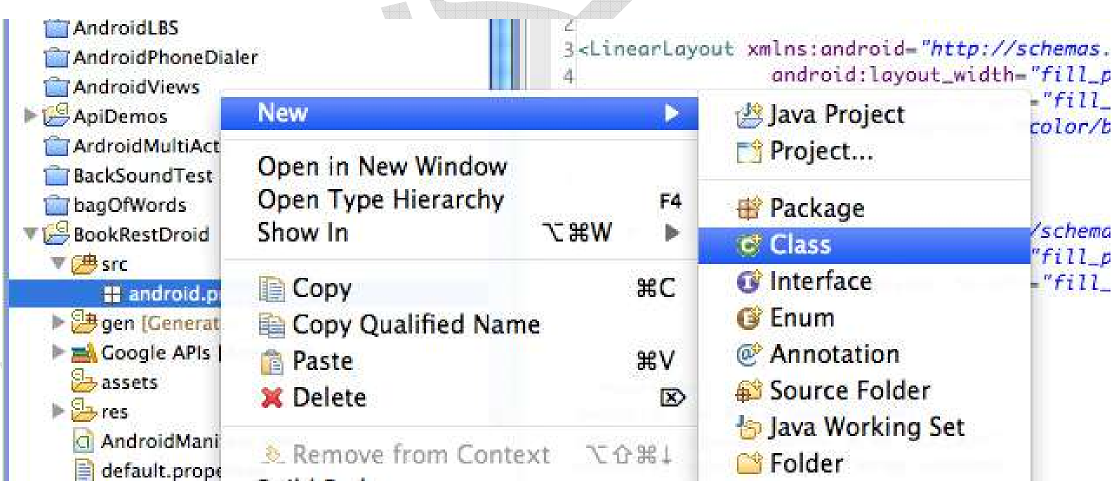

New Android Project
New Android Project
Creates a new Android Project resource.
Project name: BookRestDroid
Contents
Create new project in workspace
Create project from existing source
Use default location
Location: /Users/wa2el/Documents/workspace/BookRestDroid
Browse...
Create project from existing sample
Samples: MapsDemo
Build Target
Target Name
Vendor
Platform
AI
Android 1.5 | Android Open Source Project | 1.5 | 3
Google APIs | Google Inc. | 1.5 | 3
Android 2.0.1 | Android Open Source Project | 2.0.1 | 6
Google APIs | Google Inc. | 2.0.1 | 6
Android 2.1 | Android Open Source Project | 2.1 | 7
Google APIs | Google Inc. | 2.1 | 7
Android 2.2 | Android Open Source Project | 2.2 | 8
Google APIs | Google Inc. | 2.2 | 8
Standard Android platform 2.1
Properties
Application name: BookRestDroid
< Back
Next >
Cancel
Finish



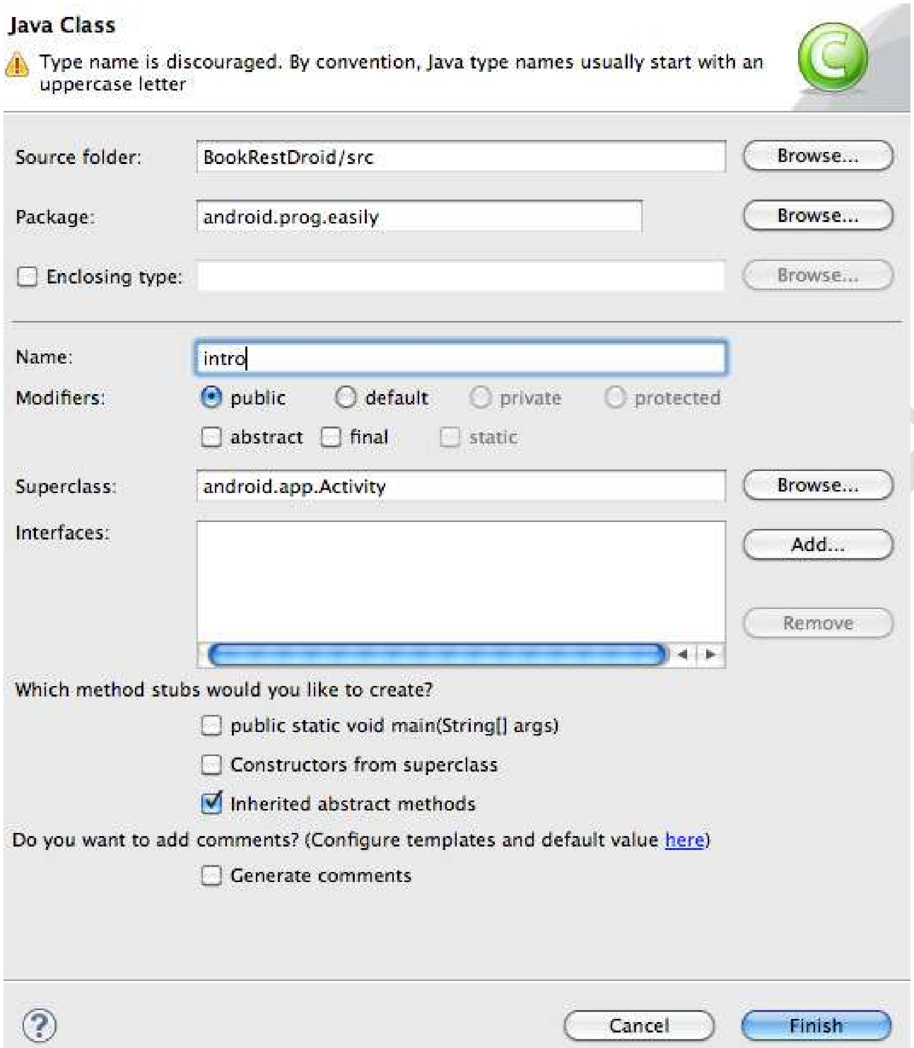
حتى نتمكن من استخدام الأحرف العربية داخل المشروع، يتوجب علينا تعريف الترميز ليكون UTF-8 في المشروع. اضغط بالزر الأيمن على اسم المشروع ومن ثم:Properties



نقوم الآن بإنشاء مجموعة Class بمسمى:intro

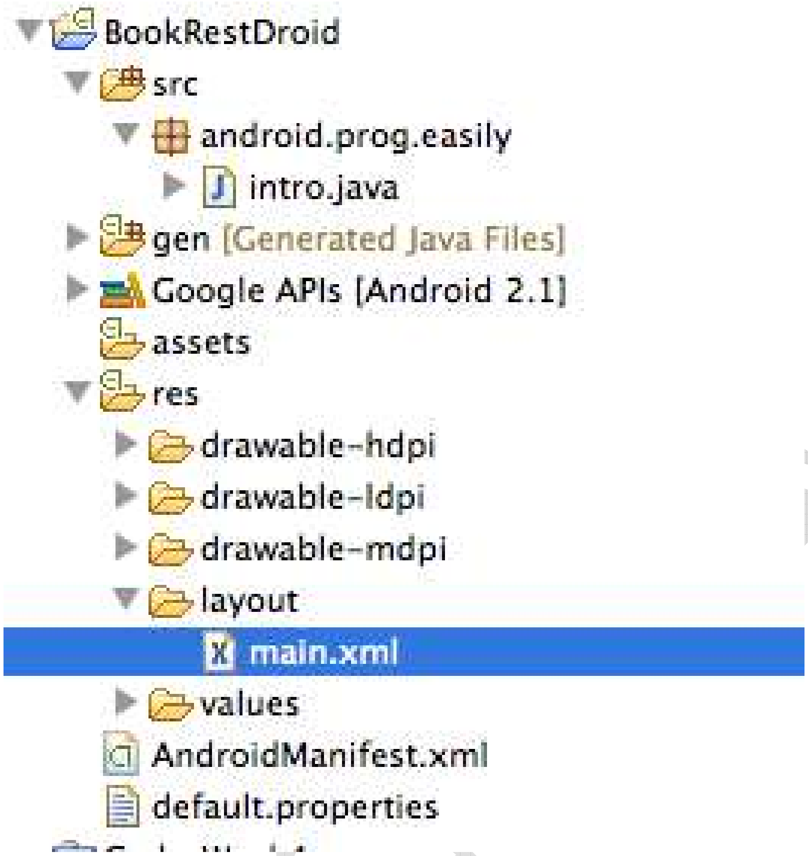


ونملأ البيانات الخاصة بها:



(لاحظ اسم المجموعة التي تعلو المجموعة المنشأة) android.app.Activity

قبل الشروع في كتابة الكود البرمجي الخاص بهذه المجموعة (والتي ستقترن بالواجهة الإبتدائية التي ستظهر للمستخدم)، يفضل أن نقوم بإنشاء الواجهة المرئية. لعمل ذلك، نقوم باستخدام ملف main.xml الذي تم إنشاؤه تلقائيا عند إنشاء المشروع.



الواجهة التي نريد تصميمها بسيطة جدا، وستكون على الشكل التالي:
خلفية بيضاء، صورة تظهر في المنتصف، زر.

بالنسبة للخلفية البيضاء، سنترك ملف main.xml قليلا ونقوم بإنشاء ملف نسميه مثلا colors.xml ونضعه تحت مجلد values (يتم الإنشاء بالضغط بالزر الأيمن على مجلد values ومن ثم new>file>colors.xml تسمية ملفات الاكس ام ال يجب أن تتم باستخدام حروف صغيرة lower case وعليها أن تخلو من الرموز.)
محتوى الملف:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<color name="background">#FFFFFFFF</color>
</resources>
```

هذا النوع من الملفات له دور مهم في عملية التطوير، فباستخدام أسماء المتغيرات الموجودة بداخله في الكود البرمجي، يمكنك فيما بعد مجرد تعديل قيم المتغيرات في هذا الملف ليتم استخدام القيمة الجديدة في البرنامج ككل. وهذا الأمر ينطبق على ملف strings.xml في نفس المجلد وفيه يتم تعريف متغيرات نصية يتم الإشارة إليها بالاسم في الكود، وعندما يراد تغيير قيم هذه النصوص، فبدلا من البحث عن كل متغير في الكود البرمجي، نلجأ إلى هذا الملف تحديدا ونغير القيم الموجودة بداخله، **ولكنني للأسف لا أتبع هذه الطريقة** حيث أقوم بكتابة النصوص داخل الكود، فمعذرة.

نعود الآن إلى ملف الـmain.xml ، ونضيف التالي:

```
<!-- هذا السطر يجب أن يكون متواجدا وهو يعرف النسخة المستخدمة ونوع الترميز -->
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:background="@color/background">


<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent">

<!-- يستخدم لتعريف الصورة وتحديد مصدرها-->

<ImageView
```

```
android:id="@+id/ImageView01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:src="@drawable/andbooklogo"
android:adjustViewBounds="true"
android:layout_centerHorizontal="true"
>
</ImageView>

<!-- يستخدم لتعريف الزر -->

<Button android:id="@+id/search"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/ImageView01"
android:paddingLeft="20px"
android:paddingRight="20px"
android:layout_centerInParent="true"
android:layout_marginTop="20px"
/>

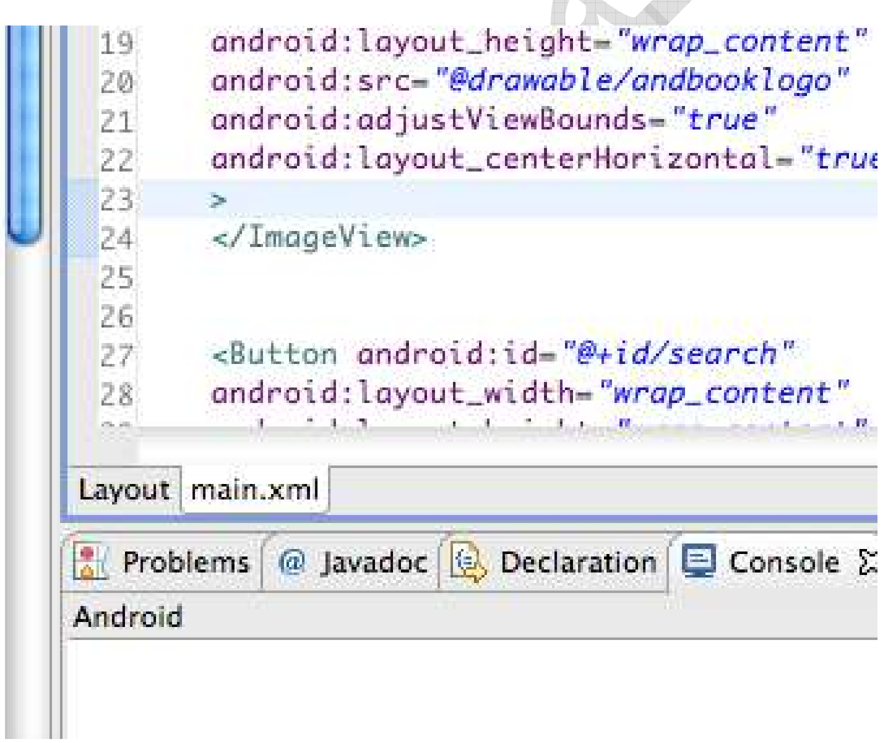
</RelativeLayout>
</LinearLayout>
```

نلاحظ أننا في ImageView قمنا بإضافة مصدر الصورة التي نريد استخدامها. لإضافة الصورة إلى المشروع، نقوم بالضغط عليها وسحبها إلى مجلد res/drawable-hdpi أوres/drawable بحسب الموجود (يمكنكم استخدام صورة لديكم أو تلك الموجودة في ملفات التطبيق على الموقع).

السؤال القادم هو ماذا عن LinearLayout وRelativeLayout ؟ للتعرف عليها، أدعوكم لقراءة الملحق "أ" من هذا الكتاب.

لم أقم بشرح RelativeLayout كونه يتشابه مع الأوعية الباقية في المبدأ ولكنه يختلف عنهم في أن ترتيب العناصر (الأزرار والصور وغيرها) يكون بنسبة أماكنها بين بعضها. بمعنى اذا رغبت بوضع زر أسفل الصورة مع محاذاته الى اليمين، فهذا الوعاء سيعطيك هذه الإمكانية حيث ستعرف الزر بأنه سيقع أسفل الصورة مع محاذاته إلى اليمين. وهذا تفسير android:layout_below في تعريف الزر، والقيمة المعطاة هي **رقم معرف** الصورة.

انتهينا من عمل الواجهة المرئية، ويمكننا استعراضها بالضغط على زر Layoutكما في الشكل:



والآن نعود لملف المجموعة . intro الكود سيكون التالي: (سأقوم بالشرح على الكود)

```
package android.prog.easily;
```

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class intro extends Activity
{
//تعريف متغير من نوع زر
private Button getbest50;

@Override
   public void onCreate(Bundle savedInstanceState)
{
      super.onCreate(savedInstanceState);

//        نستخدم ملف main من أجل الواجهة المرئية
      setContentView(R.layout.main);
      setUpViews();
   }

private void setUpViews()
{
getbest50 = (Button)findViewById(R.id.search);
getbest50.setText("أفضل ٥٠ مطعم");

//تعريف ماذا سيتم في حال الضغط على الزر
getbest50.setOnClickListener(new
View.OnClickListener()
{
public void onClick(View v)
```

```
{
top50Handler();
}
});

}

protected void top50Handler()
{

}
}
```

ما يتوجب علينا عمله في حال الضغط على الزر هو التالي:
الاتصال بالانترنت وجلب أفضل ٥٠ مطعما في مدينة الرياض، ومن ثم الانتقال إلى واجهة تعرض المطاعم وتقييماتها .

سنتوقف هنا قليلا ونترك ملف intro.java والذي يمكنك بالمناسبة تشغيله على المحاكي ورؤية شكله الحالي، ولن يحصل شيئا إذا ضغطنا على الزر) ونقوم بتعريف المجموعة RatedItem.java بالشكل التالي:

```
package android.prog.easily;

import java.io.Serializable;

public class RatedItem implements Serializable
{
private static final long serialVersionUID =
5500258407135652423L;
```

```
private String name;
private int id;
private int voteCount;
private double wScore;

public RatedItem(String name, int id, int voteCount,
double wScore)
{
this.name = name;
this.id = id;
this.voteCount = voteCount;
this.wScore = wScore;
}

public String getName()
{
return name;
}

public void setName(String name)
{
this.name = name;
}

public int getId()
{
return id;
}

public void setId(int id)
{
```

```
this.id = id;
}

public int getVoteCount()
{
return voteCount;
}

public void setVoteCount(int voteCount)
{
this.voteCount = voteCount;
}

public double getwScore()
{
return wScore;
}

public void setwScore(int wScore)
{
this.wScore = wScore;
}

}
```


والآن سنقوم بإنشاء ملف RestDroidApplication.java والذي كما قلنا سيحتوي على دوال ومتغيرات يمكن لجميع الواجهات والأنشطة Activities استخدامها (أي لا داعي لنسخ الدوال نفسها في كل الأنشطة عدة مرات).

المجموعة تحتوي ما يلي:

```
package android.prog.easily;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;
import android.app.Application;

public class RestDroidApplication extends Application
{

 //يتوجب الحصول عليه من موقع قيم Qaym API Key
   private String key = "xxxxxxxxxxxxxxxxxxxxx";
   private int[] countryAndCityIds;
   private JSONObject[] ratedRestaurants;
   private ArrayList<RatedItem> globalAdapterData;
   private RatedItem itemOnMap;
   private String branchesDetailsText;

public String getBranchesDetailsText()
```

```
{
return branchesDetailsText;
}

public void setBranchesDetailsText(String
branchesDetailsText)
{
this.branchesDetailsText = branchesDetailsText;
}

public RatedItem getItemOnMap()
{
return itemOnMap;
}

public void setItemOnMap(RatedItem itemOnMap)
{
this.itemOnMap = itemOnMap;
}

@Override
public void onCreate()
{
super.onCreate();
}

@Override
public void onTerminate()
{
super.onTerminate();
}
```

```
//دالة تقوم بعمل طلب HTTP للحصول من معلومات الموقع المعطى
 عبر بروتوكولREST
   public String queryRESTurl(String url)
   {
HttpClient httpclient = new DefaultHttpClient();

HttpGet httpget = new HttpGet(url);

HttpResponse response;

try
{
response = httpclient.execute(httpget);

HttpEntity entity = response.getEntity();

if (entity != null)
{
InputStream instream = entity.getContent();
String result1 =
convertStreamToString(instream);

instream.close();

return result1;
}

}

catch (IOException e)
```

```
{

}

return null;

}
//دالة تقوم بتحويل المعلومات المستلمة إلى صيغة String يمكن
التعامل معها
   public String convertStreamToString(InputStream is)
throws IOException
   {

      if (is != null)
      {
         StringBuilder sb = new StringBuilder();
         String line;

         try
         {
            BufferedReader reader = new BufferedReader(new
            while ((line = reader.readLine()) != null)
            {
               sb.append(line).append("\n");
            }
         }

         finally
         {
            is.close();
         }
```

```
            return sb.toString();
        }

        else
        {
            return "";
        }
    }
    /*
المعلومات المستلمة من قيم تتبع الترميز JSON ولكن المكتبة البرمجية
الموجودة بالجافا والتي تحمل نفس اسم الترميز تتعامل مع الترميز بشكل مختلف
نوعا ما لذلك اضطررت لعمل بعض التغييرات على القيم المستلمة لأتمكن من
استخلاصها وتعبئتها في مصفوفة من عناصر JSON التي يسهل التعامل معها
لأنها تعمل على نمط مفتاح وقيمة .. Key/value يمكنكم تتبع السياق البرمجي
للكود لمعرفة الطريقة التي اتبعتها (وهي بسيطة).. كما يلزمكم مراجعة صفحة
توثيق "قيم" ورؤية
الصيغة التي يتم فيها إرسال البيانات ومقارنتها بالصيغة التي تقبلها الجافا
 لعناصر الـJSON
    */
    public JSONObject[] retrieveJSONs(String urlString)
    {
        String result = queryRESTurl(urlString);
        JSONObject[] jarr = null;

        if (result != null)
        {
        try
        {
          result = result.substring(1, result.length()-
1);
```

```
        int counter = 0;
        boolean firstSeen = false;
        boolean secondSeen = false;

// حلقة من نوع for لمعرفة عدد العناصر المستلمة وبالتالي تحديد عدد عناصر JSON المطلوبة

        for (int i = 0; i < result.length(); i++)
        {
   if (result.charAt(i) == '{' && firstSeen ==
  {
  firstSeen = true;
  counter++;
  }

  else if (result.charAt(i) == '{' && firstSeen
true)                                       ==
  {
  secondSeen = true;
  }

  if (result.charAt(i) == '}' && secondSeen
true)                                       ==
  {
  secondSeen = false;
  }

  else if (result.charAt(i) == '}' &&
false)                                secondSeen ==
  {
  firstSeen = false;
```

```
  }

        }

        jarr = new JSONObject [counter];

        int ind1 = 0;
        int ind2 = 0;

        boolean firstSeenS = false;
        boolean secondSeenS = false;

 JSONحلقة لإنشاء عناصر   //
        for (int i = 0; i < result.length(); i++)
        {
  firstSeenS = false;
        secondSeenS = false;

  if (result.charAt(i) == '{' && firstSeenS
false)                              ==
  {
  firstSeenS = true;
  ind1 = i;
  for (int j = i+1; j < result.length();
j++)
  {
    if (result.charAt(j) == '{' &&
  {
  secondSeenS = true;
  }
```

```
    if (result.charAt(j) == '}' &&
  {
  secondSeenS = false;
  }

  else if (result.charAt(j) == '}'
                                                  &&
secondSeenS == false)
  {
                    String mid =

              jarr[ind2] = new
  ind2++;
  i = j;
  break;
  }
  }
  }
        }
      }
      catch (JSONException e)
      {

   }
      }

      return jarr;
   }

//    دالة تقوم بعمل النص الكامل لطلب ال API وتقوم بإضافة المفتاح له
public String getUrlStringCommand(String command)
```

```
    {
String service_url = "http://api.qaym.com/0.1/";
String url = "";
        url = service_url + command + "/key=" + key;

        return url;
    }

public void setCountryAndCityIds(int[]
countryAndCityIds)
{
this.countryAndCityIds = countryAndCityIds;
}

public int[] getCountryAndCityIds()
{
return countryAndCityIds;
}

public JSONObject[] getRatedRestaurants()
{
return ratedRestaurants;
}

public void setRatedRestaurants(JSONObject[]
ratedRestaurants)
{
this.ratedRestaurants = ratedRestaurants;
}

public ArrayList<RatedItem> getGlobalAdapterData()
```

```
{
return globalAdapterData;
}

public void
        setGlobalAdapterData(ArrayList<RatedItem>
globalAdapter)
{
this.globalAdapterData = globalAdapter;
      }

public RatedItem getGlobalAdapterDataItem(int
position)
{
return globalAdapterData.get(position);
}
}
```

بالعودة لملف intro.java ، نقوم بإضافة الكود التالي:

```
protected void top50Handler()
{
try
{
//عمل نص ال API الخاص بجلب أفضل ٥٠ مطعما
من مدينة الرياض "الكود الخاص بها ٥٧"
String attrcommand = "cities/" + "57" +
"/items/top";
```

```
//بمجرد استخدام دالة يسبقها
accessApplicationMethods()اعرفوا أننا نقوم باستخدام دالة من ملف
RestDroidApplication.java

String attributesCommand =

accessApplicationMethods().getUrlStringCommand(attrcom
mand);

   JSONObject[] topCities =

accessApplicationMethods().retrieveJSONs(attributesComma
nd);

//مصفوفة سنقوم بتعبئتها بعناصر من نوع
ReatedItemوتزويد كل عنصر بمعلوماته الخاصة بالاستعانة بكائنات\عناصر
JSONالتي قمنا بتكوينها في السطر السابق
ArrayList<RatedItem> sRest = new
ArrayList<RatedItem>();

if (topCities != null)
{
for (int ff = 0; ff < topCities.length;
ff++)
{
//لاحظوا كيفية الحصول على قيمة من
عنصر JSON عبر مفتاح  "item_id" وقس على ذلك
int id =
Integer.parseInt(topCities[ff].getString("item_id"));
```

```
String name =
topCities[ff].getString("item_name");

int voteConutt =
Integer.parseInt(topCities[ff].getString("total_number_of_vot
es"));

double weScore =
Double.parseDouble(topCities[ff].getString("weighted_score
"));

//تعديل العلامة الموزونة لتكون على
مقياس ٥ بدلا من ١٠
double weScoreR = weScore *
100;
weScoreR =
Math.round(weScoreR);
weScoreR = weScoreR / 10 / 2;

RatedItem singleRest = new
RatedItem(name, id,
voteConutt, weScoreR);
sRest.add(singleRest);
}
}
else
{
RatedItem singleRest = new
RatedItem("Empty List", 0, 0, 0);
sRest.add(singleRest);
}
```

```
accessApplicationMethods().setGlobalAdapterData(sRest);
// ***      goToBest50Act();
}
catch (JSONException e)
{

        }
}

//دالة يتم تعريفها في أي نشاط يستخدم دوال المجموعة
RestDroidApplication.javaونتمكن بها من دخول الدوال الموجودة بها
protected RestDroidApplication
accessApplicationMethods()
{
return (RestDroidApplication)getApplication();
}
```

الإضافة السابقة تمكننا من جلب أفضل ٥٠ مطعما. وبقي لدينا الانتقال للواجهة التي تعرض المطاعم، وهذا ما يقوم به السطر الذي رمزت له بـ :

// ***

والذي يستدعي الدالة goToBest50Act ونعرفها بالشكل التالي:

```
protected void goToBest50Act()
{
Intent goToBest = new Intent(this, Best50.class);
startActivity(goToBest);
}
```

ليصبح ملف intro.java بالشكل التالي:

```
package android.prog.easily;

import java.util.ArrayList;

import org.json.JSONException;
import org.json.JSONObject;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class intro extends Activity
{
//تعريف متغير من نوع زر
private Button getbest50;

@Override
  public void onCreate(Bundle savedInstanceState)
{
      super.onCreate(savedInstanceState);
//          نستخدم ملف main من أجل الواجهة المرئية
      setContentView(R.layout.main);
      setUpViews();
   }

private void setUpViews()
```

```
{
getbest50 = (Button)findViewById(R.id.search);
xt("

//تعريف ماذا سيتم في حال الضغط على الزر
getbest50.setOnClickListener(new
View.OnClickListener()
{
public void onClick(View v)
{
top50Handler();
}
});

}

protected void goToBest50Act()
{
Intent goToBest = new Intent(this, Best50.class);
startActivity(goToBest);
}

protected void top50Handler()
{
try
{
//عمل نص ال API الخاص بجلب أفضل ٥٠ مطعما
من مدينة الرياض "الكود الخاص بها ٥٧"
String attrcommand = "cities/" + "57" +
"/items/top";
```

```
//بمجرد استخدام دالة يسبقها
accessApplicationMethods()اعرفوا أننا نقوم باستخدام دالة من ملف
RestDroidApplication.java
String attributesCommand =
accessApplicationMethods().getUrlStringCommand(attrcom
mand);

JSONObject[] topCities =
accessApplicationMethods().retrieveJSONs(attributesComma
nd);

//مصفوفة سنقوم بتعبئتها بعناصر من نوع
ReatedItemوتزويد كل عنصر بمعلوماته الخاصة بالاستعانة بكائنات\عناصر
JSONالتي قمنا بتكوينها في السطر السابق
ArrayList<RatedItem> sRest = new
ArrayList<RatedItem>();

if (topCities != null)
{
for (int ff = 0; ff < topCities.length;
ff++)
{
//لاحظوا كيفية الحصول على قيمة من
عنصر JSON عبر مفتاح  "item_id" وقس على ذلك
int id =
Integer.parseInt(topCities[ff].getString("item_id"));

String name =
topCities[ff].getString("item_name");

int voteConutt =
```

```
Integer.parseInt(topCities[ff].getString("total_number_of_vot
es"));

double weScore =
Double.parseDouble(topCities[ff].getString("weighted_score
"));

//تعديل العلامة الموزونة لتكون على
مقياس ٥ بدلا من ١٠
double weScoreR = weScore *
100;
weScoreR =
Math.round(weScoreR);
weScoreR = weScoreR / 10 / 2;

RatedItem singleRest = new
RatedItem(name, id,
voteConutt, weScoreR);
sRest.add(singleRest);
}
}
else
{
RatedItem singleRest = new
RatedItem("Empty List", 0, 0, 0);
sRest.add(singleRest);
}

//لاحظوا أننا نقوم بتمرير قائمة المطاعم إلى متغير
موجود في RestDroidApplication.java لنتمكن من التعامل معه وقراءة
محتواه في مجموعات وأنشطة Activities أخرى
```

```
accessApplicationMethods().setGlobalAdapterData(sRest);
goToBest50Act();
}
catch (JSONException e)
{

      }
}

//دالة يتم تعريفها في أي نشاط يستخدم دوال المجموعة
RestDroidApplication.javaلنتمكن من دخول الدوال والمتغيرات
الموجودة بها كما هو في الأعلى
protected RestDroidApplication
accessApplicationMethods()
{
return (RestDroidApplication)getApplication();
}

}
```

نلاحظ من الكود في الأعلى أننا قمنا بتمرير بعض البيانات إلى متغيرات موجودة في مجموعة
RestDroidApplications
والسبب كما قلنا هو لتمكين المجموعات والأنشطة الأخرى من دخول هذه المتغيرات. هناك طريقة أخرى لا أقوم بها في هذا الشرح على أنني أستخدمها حاليا بشكل أساسي ويمكن للقارئ إتباعها إذا وجدها أفضل وهي إنشاء كائن من مجموعة Intent (والتي تعرف بأنها مجموعة خاصة بـ فعل action نود عمله) وبرمجته بحيث يقوم بتمرير بيانات معينة من النشاط الحالي إلى آخر.

بالطبع ستظهر لنا رسالة خطأ كوننا لم ننشئ ملف المجموعة التي سننتقل إليها والتي سنسميها .Best50.java ولكن قبل إنشائها، يتوجب علينا إنشاء المجموعة المساعدة لها وهي:

RatedRestaurantListItem.java

ويقترن بها rest_list_item.xml وكما ذكرنا من قبل فإن هذه المجموعة تقوم بتشكيل الواجهة الخاصة بكل عنصر (مطعم) -الشكل ١- وبما أنها مرتبطة ومتعلقة بالواجهات، فستقوم هذه المجموعة بوراثة قالب\مجموعة RelativeLayoutللتعديل عليه بالشكل الذي نريده. هذا الأمر يعني أننا لن نقوم بوضع وسم RelativeLayout في ملف الـ xml المقترن، وإنما اسم مجموعتنا الجديدة -سأوضح ذلك بعد قليل.-

ننشئ مجموعة RatedRestaurantListItem.java ويكون محتواها:

```
package android.prog.easily;

import android.content.Context;
import android.util.AttributeSet;
import android.widget.RelativeLayout;

public class RatedRestaurantListItem extends RelativeLayout
{

public RatedRestaurantListItem(Context context,
AttributeSet attr)
{
super(context, attr);
}
}
```

والآن ندعها قليلا ونقوم بإنشاء ملف الـ rest_list_item.xml داخل المجلد :

أنشئ ملف جديد بالضغط على مجلد layout باليمين ومن ثم (res>layout new>file). المحتوى الخاص بهذا الملف هو ما يلي (ولاحظ(ـي) وسم القالب):

```
<?xml version="1.0" encoding="utf-8"?>
<android.prog.easily.RatedRestaurantListItem

xmlns:android="http://schemas.android.com/apk/res/android"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            >
<!-- عنصر يمكن المستخدم من الضغط على الواجهة وعمل شيء في المقابل --
>
<CheckedTextView
android:layout_width="wrap_content"
android:id="@+id/ctext"
android:layout_height="wrap_content"
android:textSize="30sp"
>
</CheckedTextView>

<!-- عنصر لعرض التقييم عليه، واستخدمنا ستايل معرف مسبقا في الأندرويد
وهو سطر من النجوم وحددناها لتكون ٥ نجوم -->
<RatingBar
android:id="@+id/rate"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/ctext"
android:numStars="5"
style="?android:attr/ratingBarStyleIndicator"
/>
```

```
<!-- واجهات نصية لعرض العلامات التي حصل عليها المطعم -->
<TextView
android:id="@+id/ratedlabel"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/ctext"
android:textSize="20sp"
android:layout_toRightOf="@id/rate"
android:paddingLeft="30px"
/>


<TextView
android:id="@+id/scorelabel"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/rate"
android:textSize="20sp"
android:paddingTop="10px"
android:paddingRight="10px"
/>

</android.prog.easily.RatedRestaurantListItem>
```


والآن نعود لملف RatedRestaurantListItem.java ونقوم باستخدام عناصر الواجهة التي عرفناها منذ لحظات ونربطها بالقيم التي نريدها عرضه (وهي معلومات المطعم). الكود يصبح:

```
package android.prog.easily;

import android.prog.easily.R;
import android.content.Context;
import android.util.AttributeSet;
import android.widget.CheckedTextView;
import android.widget.RelativeLayout;
import android.widget.TextView;

public class RatedRestaurantListItem extends RelativeLayout
{
private CheckedTextView ratedButton;
private RatedItem ratedItem1;
private TextView texty;
private TextView texty2;

public RatedRestaurantListItem(Context context,
AttributeSet attr)
{
super(context, attr);
}

//دالة يتم استدعاؤها تلقائيا بعد قراءة العناصر التي عرفناها في ملف ال
xmlوعندها يمكننا التعامل مع هذه العناصر
@Override
protected void onFinishInflate()
{
super.onFinishInflate();
ratedButton =
(CheckedTextView)findViewById(R.id.ctext);
```

```
texty =
(TextView)findViewById(R.id.ratedlabel);
texty2 =
(TextView)findViewById(R.id.scorelabel);
}

//دالة تقوم بتعيين معلومات وقيم العنصر الذي تم تمريره لها داخل
عناصر الواجهة
public void setRatedItem(RatedItem ratedItem1)
{
this.ratedItem1 = ratedItem1;
ratedButton.setText(ratedItem1.getName());
texty.setText("عدد الأصوات :
"+ratedItem1.getVoteCount());
texty2.setText("العلامة : "+ratedItem1.getwScore()
+ "/5");
}

public RatedItem getRatedItem()
{
return ratedItem1;
}
}
```

بالنسبة لـBest50.java ، وكما تعودنا، سننشئ الواجهة الخاصة بها أولا وهي : rated_rest_list.xml
وستكون بالشكل التالي:

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent">

<TextView android:id="@+id/RRselect"
android:layout_width="fill_parent"
android:layout_height="wrap_content"/>

        <ListView
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:id="@id/android:list"
                android:layout_below="@id/RRselect"/>

</RelativeLayout>
```

الآن جاء دور المجموعة .Best50.java هذه المجموعة ستقوم بعرض بيانات أفضل ٥٠ مطعما في الرياض. وبما أنها ستكون على شكل قائمة، فمن الأفضل استخدام مجموعة جزئية\خاصة من مجموعة Activity التي استخدمناها في intro.java.هذه **المجموعة** هي ListActivity وأهم ما يميزها أن القائمة Listالتي تتبناها تتطلب استخدام أحد المفاهيم المعروفة لدى مهندسي البرمجيات وهو مفهوم الوسيط Adapter أحد أنواع أنماط التصميم . Design Patterns مهمة الوسيط هو التعامل مع البيانات (المطاعم) وإنشاء الواجهات الخاصة بها، ومن ثم تمريرها لقائمة الواجهة الرئيسية التي ستحتضن هذه المعلومات (أي الواجهة المقترنة بمجموعة .(Best50 أحد الأهداف هو تقليل تداخل مجموعة Best50وواجهتها مع البيانات نفسها، فلو أردنا تغيير البيانات أو إضافة خصائص جديدة عليها، سنقوم حينها بالتعديل على الوسيط فقط، ولن نضطر إلى تغيير كود مجموعة .Best50.java تخيل لو أن هذه البيانات تستخدمها ١٠ مجموعات، وبشكل وطلبات مختلفة، هل سيكون حينها من الأسهل التعديل على الوسيط فقط (والذي تتخاطب معه المجموعات العشر كلها) أم تعديل المجموعات

كلها في حال عدم استخدام الوسيط!

الكود الخاص بالوسيط سيكون على شكل مجموعة داخلية inner class ولكن يمكنكم عمله في مجموعة منفصلة.

كود هذه المجموعة على الشكل التالي:

```
package android.prog.easily;

import java.util.ArrayList;

import android.app.Activity;
import android.app.ListActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.RatingBar;
import android.widget.TextView;

public class Best50 extends ListActivity
{
private TextView selectAttr;
private RatedRestaurantListAdapter adapter;

@Override
   public void onCreate(Bundle savedInstanceState)
{
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.rated_rest_list);
        setUpViews();
}

//دالة تقوم بالفعل المناسب في حال تم الضغط على أحد القيم المعروضة
على الشاشة
@Override
protected void onListItemClick(ListView l, View v, int
position, long id)
{
super.onListItemClick(l, v, position, id);

RatedItem ite =
accessApplicationMethods().getGlobalAdapterDataItem(posi
tion);

accessApplicationMethods().setItemOnMap(ite);

goToViewOnMapActivity();

}

private void goToViewOnMapActivity()
{
         Intent goToMapActivity = new Intent(this,

startActivity(goToMapActivity);
}

private void setUpViews()
```

```
{
selectAttr =
(TextView)findViewById(R.id.RRselect);
selectAttr.setText("اضغط على أحد المطاعم");

//إنشاء وسيط جديد واستخدامه ليكون على اتصال مع القائمة التي أنشأها هذا النشاط
        adapter = new RatedRestaurantListAdapter

setListAdapter(adapter);
}

//مجموعة داخلية InnerClass تعرف الوسيط Adapter الذي سيتعامل مع بيانات المطاعم ويقوم بإنشاء واجهة خاصة بكل واحد منها.
//سنستعين بمجموعة مخصصة لعمل وسيط يتعامل مع قوائم وهذه المجموعة هي ArrayAdapter
class RatedRestaurantListAdapter extends
ArrayAdapter<RatedItem>
{
Activity context;
private ArrayList<RatedItem> myRatedItems;
    RatedRestaurantListAdapter(Activity context,
{
super(context, R.layout.rest_list_item,
items);
this.context=context;
this.myRatedItems = items;
}
//دالة تقوم بإنشاء الواجهة الخاصة بعرض كل عنصر -كما في الشكل ١-
public View getView(int position, View
```

```
parent)                    convertView, ViewGroup
{

RatedRestaurantListItem row =
(RatedRestaurantListItem)convertView;
if (null == row)
{
//مهمة هذه المجموعة LayoutInflator تتعلق بأداء التطبيق وسرعته، حيث
أنها تقتطع جزء من البيانات وتنشئ واجهاتها
//بشكل يتناسب مع مساحة عرض الشاشة، أي
أنها لا تملأ ذاكرة الجهاز بعمل واجهة لكل البيانات وإنما تقوم بعملها بشكل
 تفاعلي
//أي كلما قام المستخدم بتمرير إصبعه لأسفل
 لرؤية المزيد من البيانات، يتم إنشاء الواجهات
LayoutInflater
inflater=context.getLayoutInflater();

row =
(RatedRestaurantListItem)inflater.inflate(R.layout.rest_list_it
em, null);
}


row.setRatedItem(myRatedItems.get(position));

RatingBar rateBar =
(RatingBar)row.findViewById(R.id.rate);
rateBar.setStepSize((float) 0.1);


rateBar.setRating((float)myRatedItems.get(position).getwSco
```

```
re());

return row;
}

public RatedItem getItem(int position)
{
return myRatedItems.get(position);
}

public void reload()
{
//دالة مهمة لإعلام القائمة التي تحتضن الوسيط بأن
بعض المعلومات تغيرت ويتوجب عليها إذا تحديث نفسها
notifyDataSetChanged();
}

}

protected RestDroidApplication
accessApplicationMethods()
{
return (RestDroidApplication)getApplication();
}

}
```

إذا كما هو واضح، فإن OnListItemClick هي دالة مهمتها القيام بعمل معين عند الضغط على أي عنصر في القائمة. العمل الذي نريد منها القيام به هو أولا جلب معلومات المطعم الذي تم الضغط عليه (يتم ذلك عبر معرفة رقم عنصر القائمة الذي تم الضغط عليه position وهو يكافئ رقم المطعم كما هو موجود

في المتغير الذي يحمل بيانات المطاعم. لذلك نقوم بجلب المطعم المختار ونضعه في متغير آخر لتستخدمه الخريطة) وثانيا الانتقال إلى واجهة أخرى تحوي الخريطة ومكان المطعم عليها.

جاء دور مجموعة ViewOnMapActivity.java والتي تقوم بعرض فروع المطعم المختار على الخريطة. سننشئ بداية ملف الـ xml (الواجهة المرئية) المقترن بها وهو: view_map.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent">
      <TextView
            android:id="@+id/restaurantNameMap"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
            android:textSize="30sp"
         />

      <com.google.android.maps.MapView
            android:id="@+id/map"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"

      android:layout_below="@id/restaurantNameMap"
            android:clickable="true"

      android:apiKey="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
xx" />

</RelativeLayout>
```

تلاحظون وجود android:apikey وهو رقم **يجب** عليكم الحصول عليه عن طريق موقع غوغل. يتم توليد الرقم باستخدام الوثيقة الالكترونية التي يتم بها توقيع التطبيقات (تطبيقات أندرويد يتم توقيعها بوثيقة الكترونية تأتي بشكل افتراضي مع الـ -SDK وهذه الوثيقة هي لدواعي التجربة debugging ويتوجب عليكم إنشاء وثيقة جديدة خاصة بكم في حال رغبتم بنشر التطبيق على الماركت). نقوم باستخراج بصمة الوثيقة الالكترونية عبر الأمر (في الوحدة الطرفية cmd أو trminal shell):

```
$ keytool -list -alias androiddebugkey \
-keystore xxx \
-storepass android -keypass android
```

نستبدل الـ xxx بأحد ما يلي بحسب نظام التشغيل:

Windows Vista: C:\Users\<user>\.android\debug.keystore

Windows XP: C:\Documents and Settings\<user>\.android\debug.keystore

OS X and Linux: ~/.android/debug.keystore

النتيجة ستكون شبيهة بما يلي:

```
Certificate fingerprint (MD5):
94:1E:43:49:87:73:BB:E6:A6:88:D7:20:F1:8E:B5:98
```

نقوم بنسخ هذا الرقم والتوجه إلى الموقع:

http://code.google.com/android/maps-api-signup.html

وإكمال الخطوات المطلوبة. وعندها سيتم تزويدنا برقم الـ API الذي نقوم بوضعه في الأعلى بدلا من الـxxxxxxxxxx.

لمزيد من المعلومات، زوروا هذه الصفحة:
http://code.google.com/android/add-ons/google-apis/mapkey.html

والآن قبل كتابة محتوى ملف ViewOnMapActivity.java ، سنقوم بكتابة كود المجموعة المساعدة:AddressOverlay.java :

```
package android.prog.easily;

import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Point;
import android.location.Address;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;
import com.google.android.maps.Projection;

public class AddressOverlay extends Overlay
{
        private static final int CONTAINER_RADIUS
private static final int
= 1;        CONTAINER_SHADOW_OFFSET
```

```
private Address address;
private GeoPoint geopoint;
Bitmap bb;

public AddressOverlay(Address address)
{
super();
assert(null != address);
//استخلاص معلومات العنوان الذي يتم تمريره
this.setAddress(address);
Double convertedLongitude =
address.getLongitude() * 1E6;
Double convertedLatitude = address.getLatitude()
* 1E6;

setGeopoint(new GeoPoint(
convertedLatitude.intValue(),
convertedLongitude.intValue()));
}

//دالة تقوم برسم نقطة خضراء فوق الخريطة عند العنوان المحدد
@Override
public void draw(Canvas canvas, MapView mapView,
boolean shadow)
{
super.draw(canvas, mapView, shadow);
Point locationPoint = new Point();
Projection projection = mapView.getProjection();
projection.toPixels(getGeopoint(), locationPoint);
Paint containerPaint = new Paint();
containerPaint.setAntiAlias(true);
```

```
int containerX = locationPoint.x;
int containerY = locationPoint.y;
if (shadow)
{
containerX +=
CONTAINER_SHADOW_OFFSET;
containerY +=
CONTAINER_SHADOW_OFFSET;
containerPaint.setARGB(90, 0, 0, 0);

canvas.drawCircle(containerX, containerY,
CONTAINER_RADIUS, containerPaint);
}

else
{
 containerPaint.setColor(Color.GREEN);
canvas.drawCircle(containerX, containerY,
CONTAINER_RADIUS, containerPaint);
}
}

public void setAddress(Address address)
{
this.address = address;
}

public Address getAddress()
{
return address;
}
```

```
public void setGeopoint(GeoPoint geopoint)
{
this.geopoint = geopoint;
}

public GeoPoint getGeopoint()
{
return geopoint;
}
}
```

والآن وأخيرا، سنقوم بكتابة كود المجموعة الأخيرة ViewOnMapActivity.java:

```
package android.prog.easily;

import java.io.IOException;
import java.util.List;

import org.json.JSONException;
import org.json.JSONObject;

import android.graphics.Bitmap;
import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import android.widget.TextView;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
```

```
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;

public class ViewOnMapActivity extends MapActivity
{
public static final String ADDRESS_RESULT =
"address";


private MapView mapView;
private Address address;
private TextView restName;
private JSONObject[] restaurantsLocObjects;
private String[] restaurantsBranchesNames;
private double [][] longLat;
private int maxLat = 0;
private int minLat = 0;
private int maxLon = 0;
private int minLon = 0;
Bitmap bmp;

@Override
protected void onCreate(Bundle bundle)
{
super.onCreate(bundle);
setContentView(R.layout.view_map);
setUpViews();

bringRestLocations();
```

```
mapCurrentAddress();
}

//دالة لجلب موقع المطعم المختار وفروعه
private void bringRestLocations()
{
try
{
//جلب رقم معرف ID المطعم الذي تم اختياره في
 المجموعة السابقة Best50.java وتم تخزينه في متغير في مجموعة
RestDroidApplication
int restNum =
accessApplicationMethods().getItemOnMap().getId();

//تشكيل نص أمر الـ API متضمنا رقم المطعم الذي
نريد معلوماته
String command = "items/" + restNum +
"/locations";

String restaurantsCommand =
accessApplicationMethods().getUrlStringCommand(comman
d);

restaurantsLocObjects =
accessApplicationMethods().retrieveJSONs(restaurantsCom
mand);

restaurantsBranchesNames = new
String[restaurantsLocObjects.length];

longLat = new double[restaurantsLocObjects.length][2];
```

```
for (int i = 0; i <
restaurantsBranchesNames.length; i++)
{
restaurantsBranchesNames[i] = (i+1)
" + البلد+ "\n" +" :
restaurantsLocObjects[i].getString("country") + "\n"
: " المدينة+ "
+
restaurantsLocObjects[i].getString("city") + "\n"
: "رقم الفرع+ "
+
restaurantsLocObjects[i].getString("location_id") + "\n"
: "العنوان+ "
+
restaurantsLocObjects[i].getString("address") + "\n"
: "رقم التلفون+ "
+
restaurantsLocObjects[i].getString("phone") + "\n"
: "اسم الفرع+ "
+
restaurantsLocObjects[i].getString("title") + "\n";

longLat[i][0] =
Double.parseDouble(restaurantsLocObjects[i].getString("latit
ude"));
longLat[i][1] =
Double.parseDouble(restaurantsLocObjects[i].getString("lon
                    gitude"));
}
```

```
}

catch (JSONException e)
        {

    }

}

//دالة تقوم بوضع فروع المطعم على الخريطة
protected void mapCurrentAddress()
{
Geocoder g = new Geocoder(this);
List<Address> addresses;

boolean firstTime = true;

for (int i = 0; i <
restaurantsBranchesNames.length; i++)
{
try
{
addresses =
g.getFromLocation(longLat[i][0], longLat[i][1], 1);
if (addresses.size() > 0)
{
address = addresses.get(0);
List<Overlay> mapOverlays =
mapView.getOverlays();
AddressOverlay addressOverlay
= new AddressOverlay(address);
```

```
mapOverlays.add(addressOverlay);

//هذه الجملة الشرطية والأسطر الأربعة
التي تليها الهدف منها هو تحديد نسبة الزوم zoom المراد عمله بحيث نرى
النقاط كلها عند عرض الخريطة
//أي لو كان لدينا فرعين الأول في
الرياض والثاني في الدمام، فالزوم سيكون بعيدا بحيث تظهر النقطتان الخضر
على الخريطة معا
if (firstTime)
{
minLat =
addressOverlay.getGeopoint().getLatitudeE6();
maxLat =
addressOverlay.getGeopoint().getLatitudeE6();
minLon =
addressOverlay.getGeopoint().getLongitudeE6();
maxLon =
addressOverlay.getGeopoint().getLongitudeE6();
firstTime = false;
}

minLat = (minLat >
addressOverlay.getGeopoint().getLatitudeE6()) ?
addressOverlay.getGeopoint().getLatitudeE6() : minLat;
maxLat = (maxLat <
addressOverlay.getGeopoint().getLatitudeE6()) ?
addressOverlay.getGeopoint().getLatitudeE6() : maxLat;

minLon = (minLon >
addressOverlay.getGeopoint().getLongitudeE6()) ?
```

```
addressOverlay.getGeopoint().getLongitudeE6() : minLon;
maxLon = (maxLon <
addressOverlay.getGeopoint().getLongitudeE6()) ?
addressOverlay.getGeopoint().getLongitudeE6() : maxLon;

mapView.invalidate();
}
else
{

}
}

catch (IOException e)
{

}
}

final MapController mapController =
mapView.getController();
GeoPoint center = new GeoPoint((maxLat +
minLat)/2,(maxLon + minLon)/2);

mapController.animateTo(center, new Runnable()
{
public void run()
{
mapController.zoomToSpan(maxLat -
minLat,maxLon - minLon);
}
```

```
            });

}

private void setUpViews()
{
restName =
(TextView)findViewById(R.id.restaurantNameMap);


restName.setText(accessApplicationMethods().getItemOnMa
p().getName());
mapView = (MapView)findViewById(R.id.map);

//تفعيل أزرار التكبير و التصغير Zoomin/out
mapView.setBuiltInZoomControls(true);
}

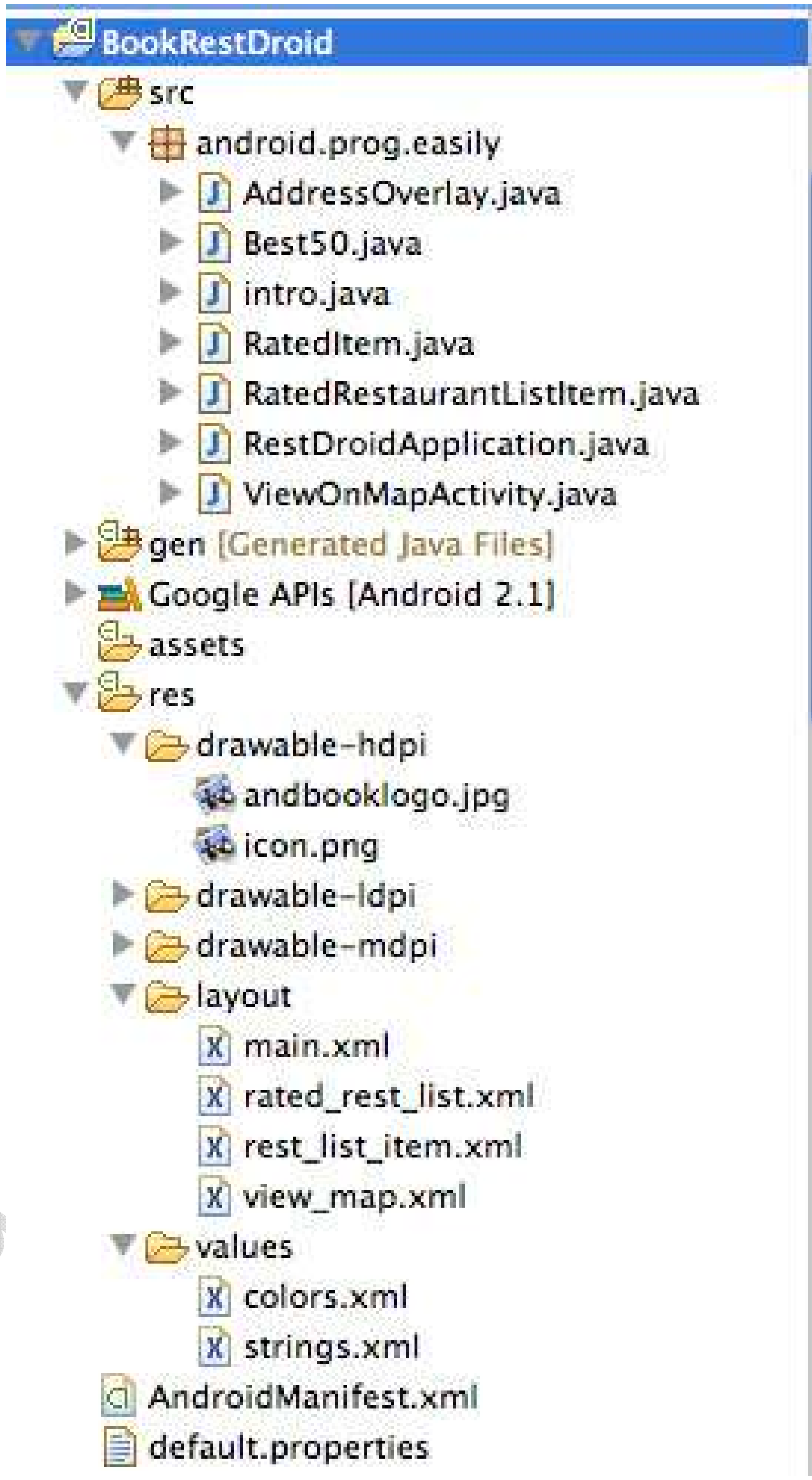
@Override
protected boolean isLocationDisplayed()
{
return true;
}

@Override
protected boolean isRouteDisplayed()
{
return false;
}

protected RestDroidApplication
```

```
accessApplicationMethods()
{
return (RestDroidApplication)getApplication();
}
}
```

والآن نلقي نظرة على ملفات المشروع:



هل نسينا شيئاً؟! الجواب **نعم**.. نسينا تعديل ملف الـ AndroidManifest.xml

وهو الملف الذي أسميه بالمايسترو أو خارطة الطريق. هذا الملف مسئول عن إعلام نظام التشغيل بكيفية تشغيل التطبيق وأي واجهة ونشاط يشغلها في البداية وغير ذلك.

نقوم بتعديله بالشكل التالي:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="android.prog.easily"
    android:versionCode="1"
    android:versionName="1.0">

    <!-- معلومات عن التطبيق تتضمن اسمه ومصدر صورة الأيقونة الخاصة به -->
    <application        android:name=".RestDroidApplication"
android:icon="@drawable/icon"
android:label="@string/app_name">

  <!-- نقوم بتعريف كل واجهة "وعددها ٣" بكتابة اسم المجموعة خاصتها بدون .java -->
     <activity android:name=".intro"
            android:label="@string/app_name">

    <!-- هذا الجزء يعني أن هذه الواجهة هي الواجهة الأولى التي نريد إطلاقها عن تشغيل التطبيق -->
        <intent-filter>
            <action
android:name="android.intent.action.MAIN" />
            <category
android:name="android.intent.category.LAUNCHER" />
```

```
        </intent-filter>


      </activity>

      <activity android:name=".Best50"
              android:label="Best 50 rest">
      </activity>

      <activity android:name=".ViewOnMapActivity"
              android:label="view on Map">
      </activity>


        <!-- نخبر نظام التشغيل أنه يتوجب عليه استدعاء مكتبة الخرائط -->
      <uses-library android:name="com.google.android.maps"
/>

   </application>

            <!-- الأذونات والتصاريح التي يمكننا بها استخدام الانترنت -->
   <uses-permission
android:name="android.permission.INTERNET" />
   <uses-permission
android:name="android.permission.ACCESS_FINE_LOCAT
ION" />

    <!-- هذا التطبيق يستخدم نسخة ٢,١ وأعلى من الأندرويد -ورقمها ٧ لدي
وهذا الرقم قد يختلف لديكم، ويمكن معرفته عند إنشاء المشروع، ستجدون رقما
بجانب النسخة- -->
   <uses-sdk android:minSdkVersion="7" />
```

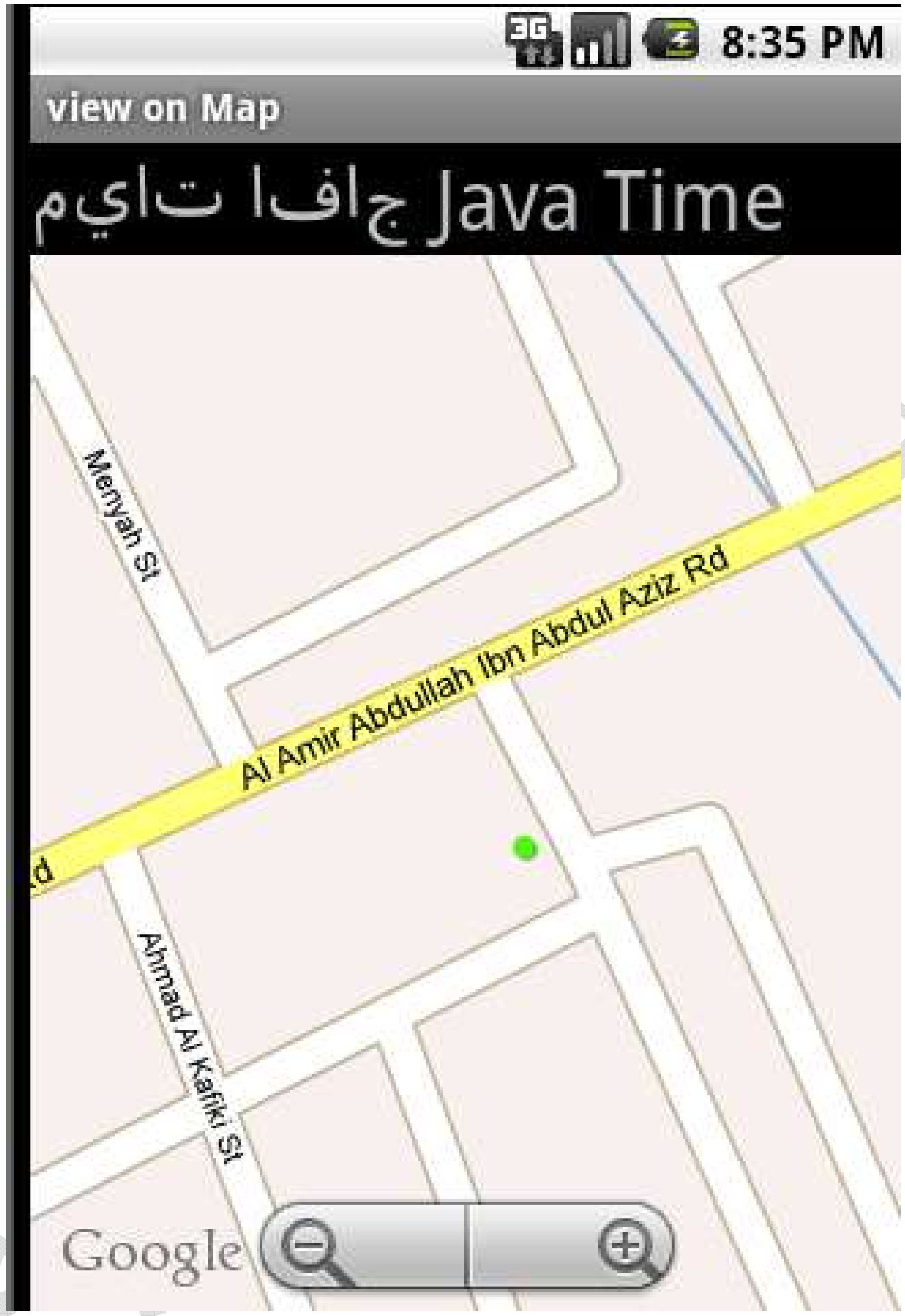
```
</manifest>
```

صور من التطبيق:




8:34 PM
Best 50 rest
اضغط على أحد المطاعم
حمص النقاء
عدد الأصوات: 52
العلامة: 4.35/5
البيتزا الشهية Delicious Pizza
عدد الأصوات: 02
العلامة: 4.2/5
اللؤلؤة الذهبية
عدد الأصوات: 03
العلامة: 4.15/5


8:34 PM
view on Map
جافا تايم Java Time
Al Wurud
King Abdul Aziz Rd
Al Takhassusi St
Ar Rahmaniyah
King Fahad Rd
Al Olaya St
Al Oruba St
As Sulaimaniyah
Moosa Bin Nusiar St
Al Amir Turki Ibn Abdul Aziz St
Al Ulayya
Al Amir Sulatan Bin Abdul Aziz St
Umm Al Hamam Algharbi
Abdul Aziz Bin Musaed Bin Julowi St
Mecca Rd
Google


8:35 PM
view on Map
جافا تايم Java Time
Menyah St
Al Amir Abdullah Ibn Abdul Aziz Rd
Ahmad Al Kafiki St
Google

متفرقات:

-التأكد من وجود أجهزة فعلية أو افتراضية (محاكي) ليتم التواصل معها عبر الوحدة الطرفية:

```
$ cd /Developer/android-sdk-mac_86/tools/
$ ./adb devices
```

النتيجة:

```
List of devices attached
emulator-5554    device
```

-مساحة الذاكرة الداخلية للجهاز الافتراضي قد تكون غير كافية للتطبيق أو لملفات نرغب برفعها إليه (كملفات الخطوط العربية)، لذلك يمكننا إطلاق محاكي بمساحة نقوم بتخصيصها عبر الأمر:

```
$ cd /Developer/android-sdk-mac_86/tools/
$ ./emulator -avd GPSenables -partition-size 500
```

اسم المحاكي يتم تحديده بعد avd- أي أن المحاكي الذي أريد تشغيله GPSenables، لذلك قم \ قومي بتغيير هذا الاسم وضع المحاكي الذي ترغب بإطلاقه.

ستنطلق نسخة من المحاكي وتجد أن الوحدة الطرفية أصبحت لا تقبل مدخلات، لذلك دع هذه النافذة وافتح نافذة وحدة طرفية أخرى.

-للقيام برفع خطوط عربية إلى المحاكي (الخطوط موجودة في الموقع الذي وضعنا عليه ملفات البرنامج) وبما أننا سنقوم بالكتابة على ملفات النظام، علينا أولا كتابة الأمر:

```
$ ./adb remount
```

النتيجة

```
remount succeeded
```

ومن ثم (مع تغيير المسار):

```
$ ./adb push /Users/wa2el/Desktop/fontss/DroidSans-Bold.ttf
/system/fonts/

$ ./adb push /Users/wa2el/Desktop/fontss/DroidSans.ttf
/system/fonts/

$ ./adb push
/Users/wa2el/Desktop/fontss/DroidSansFallback.ttf
/system/fonts/
```

-أثناء لصق الكود المنسوخ، قد تظهر لك أحيانا خطوط صفراء تحت الكود (تحصل كثيرا في ملفات الـ xml) الحل هو إعادة اللصق أعلى الكود الذي تحته خطوط صفراء ومن ثم إزالة الأخير.

الملحق "أ":

وظيفة و كيفية إنشاء ما يلي:

١ -وعاء<LinearLayout>

٢ -وعاء<TableLayout>

٣ -وعاء<ScrollView>

١ -وعاء<LinearLayout>

لدينا عادة الخصائص\العناصر التالية:

```
<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout_width="fill_parent"

android:layout_height="fill_parent"

>
```

بالنسبة لخاصية التوجه Orientation يتم عبرها تحديد طريقة رصف الأزرار وال Widgets في الصفحة. الوضع الرأسي Vertical يعني وضع العناصر بشكل عمودي، أما Horizontal فيعني صفها بشكل أفقي.

خاصية layout_width مسئولة عن تحديد عرض الوعاء، ونحن في هذه الحالة نريد من الوعاء ملأ الشاشة واحتلال جميع المساحات، لذلك نختار fill_parent.

خاصية layout_height يتم عبرها تحديد ارتفاع الوعاء، وأيضا نريده ملأ المساحات كاملة.

نقوم الآن بإنشاء زر لنرى كيفية التحكم بطريقة ظهوره في واجهة التطبيق.

```
<Button

android:text="@+id/Button01"

android:id="@+id/Button01"

android:layout_width="wrap_content"

android:layout_height="wrap_content">

</Button>
```

بالنسبة للخاصية الأولى، android:textفهي المسئولة عن تحديد النص المراد ظهوره على الزر android:id. تقوم بإعطاء رقم خاص للعنصر يميزه عن غيره. بالنسبة للخاصيتين الأخيرتين فقد تم شرحهما في الأعلى، ولكني سأتعرض للقيم التي من الممكن إعطاءها لهما. هناك ٣ احتمالات. الأول fill_parent الذي يقوم بمليء المساحة الخالية بالزر بحسب الخاصية التي أعطي لها

(layout_width أو layout_height). الثاني هو wrap_content والذي يقوم بضغط حجم الزر بحسب محتواه النصي. الاحتمال الثالث هو أن يتم تحديد الارتفاع والعرض بمقياس البيكسل.

مثال للحالة الأولى والثانية (سيملأ الزر المساحات المجاورة له بالعرض، ولكن ارتفاعه سيكون مضغوطا:)

```
<Button

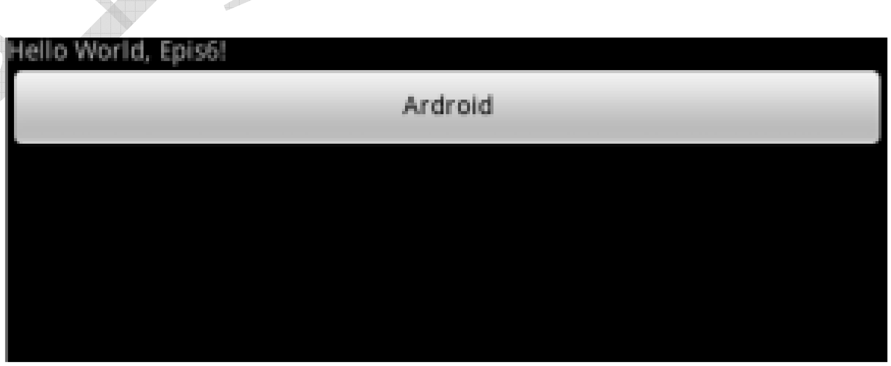
android:text="Ardroid"

android:id="@+id/Button01"

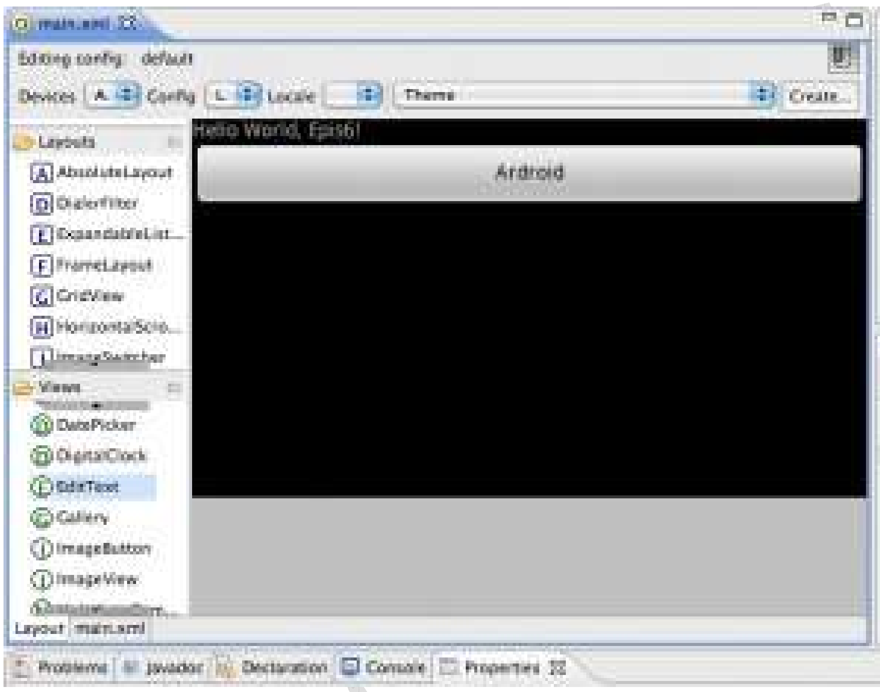
android:layout_width="fill_parent"

android:layout_height="wrap_content">

</Button>
```



)قم بتجربة الاحتمالات الأخرى، ويمكنك الاستعانة بتبويب Layout لاستعراض نتيجة التصميم بدلا من تشغيل التطبيق في كل مرة.. كما يمكنك استخدام هذه الواجهة لتصميم صفحة النشاط في التطبيق دون كتابة كود ال XML في كل مرة(

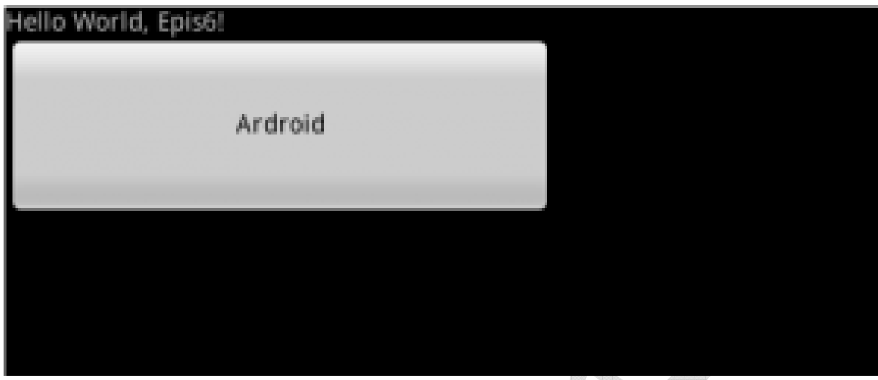


مثال للحالة الثالثة:

```
<Button
android:text="Android"
android:id="@+id/Button01"
android:layout_width="300px"
```

```
android:layout_height="100px">

</Button>
```



سنقوم بإضافة زر ثانٍ، ولكننا نريد وضعه الآن بجانب الزر الأول. كيف نقوم بذلك؟ لا ننس الخاصية الأولى الخاصة بوعاء LinearLayout وهي Orientation.نقوم بتعديلها ليتم عرض ال Widgets بشكل أفقي:

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="horizontal"

android:layout_width="fill_parent"

android:layout_height="fill_parent"

>
```

والزر الثاني (قم بتسميته بنفس اسم الزر السابق):

```
<Button

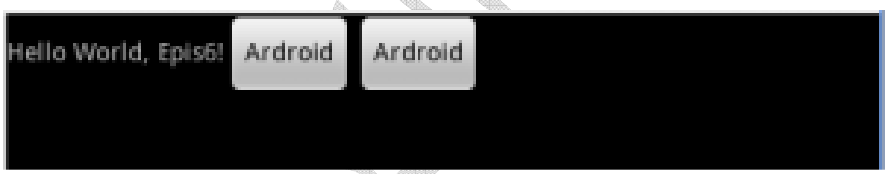
android:text="Ardroid"

android:id="@+id/Button02"

android:layout_width="wrap_content"

android:layout_height="wrap_content">

</Button>
```



سنقوم الآن بالتعرف على خاصية الوزن android:layout_weight والتي تقوم بإعطاء أولوية نقوم بتحديدها لأحد الزرين ليحتل أعلاهما أولوية كمية أكبر من المساحات الفارغة. فمثلا:

```
<Button

android:text="Ardroid"

android:id="@+id/Button01"
```

```
android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_weight="1"

>

</Button>

<Button

android:text="Ardroid"

android:id="@+id/Button02"

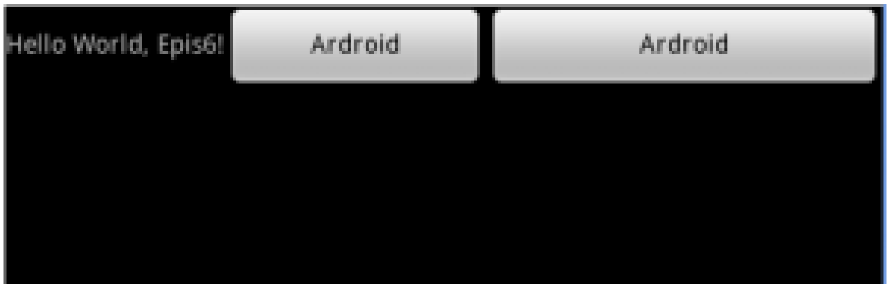
android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_weight="2"

>

</Button>
```



هناك عدد من الخصائص الأخرى والتي يتم عبرها التحكم بطريقة وضع ال Widgetsفلدينا مثلا android:layout_agravity وهي مناسبة أكثر في حالة التعامل مع ال Widgetsعند رصفهم رأسيا ونقوم عبرها بالتحكم بخاصية المحاذاة alignment التي سيكون عليها الزر مثلا. وفي حال أردنا التحكم بالمحاذاة في طريقة الرصف الأفقية، فمن الأفضل استخدام android:layout_marginLeftوتحديد عدد النقاط Pixels التي تفصل بين الزر والآخر.

مثال على android:layout_marginLeft في حالة الرصف الأفقية android:orientation=”horizontal” :

```
<Button

android:text="Ardroid"

android:id="@+id/Button02"

android:layout_width="wrap_content"

android:layout_height="wrap_content"
```

```
android:layout_marginLeft="180px"

>

</Button>
```



ومثال على android:layout_gravity بالوضع الرأسي android:orientation=”vertical”:

```
<Button

android:text="Ardroid"

android:id="@+id/Button02"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

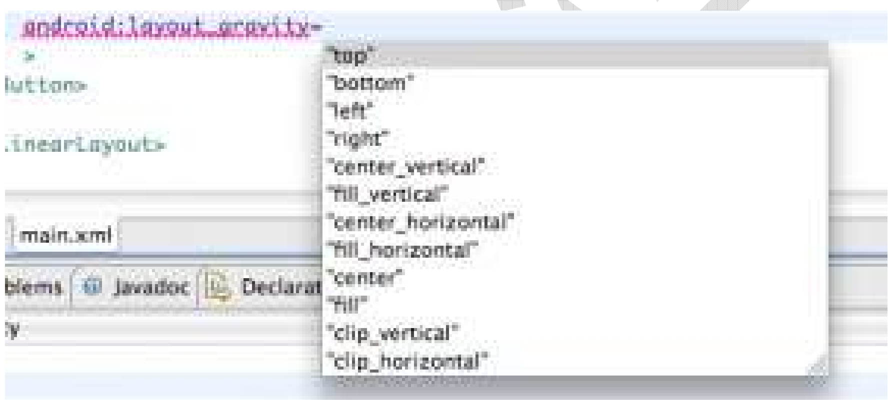
android:layout_gravity="right"

>
```

</Button>



طبعا نلاحظ القائمة المنسدلة والتي تحوي عددا من القيم التي يمكنكم تجربتها ورؤية تأثيرها.



قوموا بتجربة خصائص أخرى كال android:gravity و android:paddingLeftوغيرها ولاحظوا تأثيراتها.

في نهاية هذا القسم، تجدر الإشارة إلى أنه يمكننا تغيير خاصية توجه\وضع الوعاء Orientation أثناء عمل التطبيق. فمثلا يمكننا برمجة التطبيق بأن يقوم بتغيير التوجه ليصبح بالوضع الأفقي عند ضغط زر. تتم هذه العملية باستدعاء دالة

```
x.setOrientation(LinearLayout.HORIZONTAL);
```

حيث x هي اسم ال Widget ، زر مثلا.

٢ -وعاء<TableLayout>

يأخذ هذا الوعاء شكل الجدول. لذا، فإننا نستخدم وسم <TableRow> كلما أردنا إنشاء صف جديد. يتميز هذا الوعاء بعدد من الخصائص:

أ : android:layout_span -وهي خاصية تحدد المدى الذي نود أن يحتله العنصر، وتأخذ قيمة عددية. القيمة العددية تحدد بشكل غير مباشر عدد الأعمدة Columnsداخل الصف.

ب : android:layout_column -وهي خاصية تحدد رقم العمود الذي سيتوضع عنده العنصر .Widget وترقيم الأعمدة يبدأ من الصفر، بمعنى أن أول عمود رقمه صفر، وهكذا.

ج : android:stretchColumns -وهي خاصية محصورة بالوعاء نفسه (أي لا يمكن تطبيقها على العناصر (Widgets وهي تعمل على شد العمود ليغطي المساحة الفارغة، وتأخذ قيمة عددية تحدد رقم العمود.

د : android:shrinkColumns -وهي عكس الخاصية السابقة.

مثال توضيحي:

للحصول على التصميم التالي:



نستعمل الكود:

```
<?xml version="1.0" encoding="utf-8"?>

<TableLayout
xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout_width="fill_parent"

android:layout_height="fill_parent"
```

```
android:stretchColumns="2"

>

<TableRow>

<TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Hello!"

android:layout_column="2"

android:layout_gravity="center"

/>

</TableRow>

<TableRow>

<SeekBar

android:id="@+id/SeekBar01"
```

```
android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_span="2"

android:layout_column="1">

</SeekBar>

</TableRow>

<TableRow>

<Button

android:text="Ardroid"

android:id="@+id/Button01"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_column="2"

>
```

```
</Button>

<CheckBox

android:text="checkBox"

android:id="@+id/CheckBox01"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_column="0"

>

</CheckBox>

</TableRow>

<TableRow>

<RadioButton

android:text="Radio Button"

android:id="@+id/RadioButton01"
```

```
android:layout_width="wrap_content"

android:layout_height="wrap_content"

>

</RadioButton>

</TableRow>

</TableLayout>
```

٣ -وعاء<ScrollView>

نستعمل هذا الوعاء في حالة كان المحتوى الذي نود عرضه في الصفحة كبيرا بحيث أنه لا يمكن وضعه في صفحة واحدة. طريقة عمله سهلة وهي تضمين التصميم الذي نرغبه بداخله. فمثلا، يمكننا وضع الكود الذي كتبناه في الأعلى بداخله (بعد إضافة عدد من العناصر Widgets حتى يتطلب عرض المحتوى أكثر من صفحة).

```
<?xml version="1.0" encoding="utf-8"?>

<ScrollView

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout_width="fill_parent"
```

```
android:layout_height="wrap_content">

<TableLayout
xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout_width="fill_parent"

android:layout_height="fill_parent"

android:stretchColumns="2"

>

<TableRow>

<TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Hello!"

android:layout_column="2"
```

```
android:layout_gravity="center"

/>

</TableRow>

<TableRow>

<SeekBar

android:id="@+id/SeekBar01"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_span="2"

android:layout_column="1">

</SeekBar>

</TableRow>

<TableRow>

<Button
```

```
android:text="Android"
android:id="@+id/Button01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_column="2"
>
</Button>
<CheckBox
android:text="checkBox"
android:id="@+id/CheckBox01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_column="0"
>
```

```
</CheckBox>

</TableRow>

<TableRow>

<RadioButton

android:text="Radio Button"

android:id="@+id/RadioButton01"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

>

</RadioButton>

</TableRow>

<TableRow>

<RadioButton

android:text="Radio Button"
```

```
android:id="@+id/RadioButton01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
>
</RadioButton>
</TableRow>
<TableRow>
<RadioButton
android:text="Radio Button"
android:id="@+id/RadioButton01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
>
</RadioButton>
```

```xml
</TableRow>

<TableRow>

<TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Hello!"

android:layout_column="2"

android:layout_gravity="center"

/>

</TableRow>

<TableRow>

<TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"
```

```
android:text="Hello!"

android:layout_column="2"

android:layout_gravity="center"

/>

</TableRow>

<TableRow>

<SeekBar

android:id="@+id/SeekBar01"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_span="2"

android:layout_column="1">

</SeekBar>

</TableRow>
```

```
<TableRow>

<SeekBar

android:id="@+id/SeekBar01"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_span="2"

android:layout_column="1">

</SeekBar>

</TableRow>

<TableRow>

<SeekBar

android:id="@+id/SeekBar01"

android:layout_width="wrap_content"

android:layout_height="wrap_content"
```

```
android:layout_span="2"
android:layout_column="1">
</SeekBar>
</TableRow>
</TableLayout>
</ScrollView>
```



انتهى بحمد الله

أرجوا أن يتقبل الله هذا العمل كما أتمنى أن يكون الكتاب قد أفادكم و أرجوا منكم دعوة بظهر الغيب

محمد بدوي ، وائل علواني

دمتم بود،،،